Combining search strategies to improve performance in the calibration of economic ABMs

Aldo Glielmo[*†1], **Marco Favorito**[*‡1], **Debmallya Chanda**[2,3], **Domenico Delli Gatti**[2],

[1]Banca d'Italia, Italy[§]
[2]Università Cattolica del Sacro Cuore, Italy
[3]Universität Bielefeld, Germany

Abstract

Calibrating agent-based models (ABMs) in economics and finance typically involves a derivative-free search in a very large parameter space. In this work, we benchmark a number of search methods in the calibration of a well-known macroeconomic ABM on real data, and further assess the performance of "mixed strategies" made by combining different methods. We find that methods based on random-forest surrogates are particularly efficient, and that combining search methods generally increases performance since the biases of any single method are mitigated. Moving from these observations, we propose a reinforcement learning (RL) scheme to automatically select and combine search methods on-the-fly during a calibration run. The RL agent keeps exploiting a specific method only as long as this keeps performing well, but explores new strategies when the specific method reaches a performance plateau. The resulting RL search scheme outperforms any other method or method combination tested, and does not rely on any prior information or trial and error procedure.

1 Introduction and literature review

The last decades have witnessed a consistent growth of the reach and scope of agent-based models (ABMs) in economics and finance, certainly also as a consequence of continuing improvements in the computer hardware and software that form the foundation over which ABMs are designed and used (Axtell and Farmer 2022). ABMs have also become mature enough that they have seen adoption and usage within central banks and other financial institutions for specific tasks (Turrell 2016; Plassard et al. 2020). A particularly noteworthy application domain is the modelling of the housing market, pioneered by Bank of England (Baptista et al. 2016) and later studied by many other central banks (Cokayne 2019; Catapano et al. 2021; Carro 2022; Méro et al. 2022), and the macroeconomic model proposed in (Poledna et al. 2023) and recently adopted by Bank of Canada (Hommes et al. 2022). Other successful applications can be found in the modelling of financial stability (Bookstaber, Paddrik, and Tivnan 2014; Covi, Montagna, and Torri 2020), or of the banking sector (Chan-Lau 2017).

In spite of these success stories, ABMs are still predominantly an object of academic interest, and occupy a minor role in policy making. One fundamental reason behind ABMs' limited adoption is the overwhelming flexibility of such a modelling tool which, if handled incorrectly, can lead to widely different models of the same phenomenon and consequently to a narrow predictive power.

Rigorous calibration of ABMs via large amounts of real data is a promising path to address the problem of ABM flexibility by appropriately restricting it in data-driven and systematic manner (Axtell and Farmer 2022). In fact, ABM calibration has a long history (Fagiolo, Moneta, and Windrum 2007), but interest in ABM calibration has grown particularly in recent times of ever-increasing data abundance. Historically, the problem has been approached mostly via the 'method of simulated moments' (Gilli and Winker 2003; Franke 2009; Grazzini and Richiardi 2015), which involves minimising a measure of distance between summary statistics of real and simulated time series, while more recently, other approaches based on maximum likelihood or Bayesian statistics have been proposed and successfully tested (Grazzini, Richiardi, and Tsionas 2017; Platt 2021; Farmer et al. 2022).

A common challenge of all calibration frameworks is the need of efficiently searching for optimal parameter combinations in high-dimensional spaces, a problem made particularly arduous by the high computational cost of state-of-the-art ABM simulations. This is why the use of several heuristic search methods has been proposed in the ABM literature. Specifically, in (Lamperti, Roventini, and Sani 2018), building on the work of (Conti and O'Hagan 2010), the authors propose the use of machine surrogates, specifically in the form of XG-boost regressors, to suggest promising parameter combinations by interpolating the results of previously computed ABM simulations. In (Angione, Silverman, and Yaneske 2022), the authors expand on this idea and test the ability of several machine learning surrogate algo-

[*]These authors contributed equally.
[†]aldo.glielmo@bancaditalia.it
[‡]marco.favorito@bancaditalia.it
[§]The views and opinions expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of Banca d'Italia.

rithms such as Gaussian processes, random forests and support vector machines, to reproduce ABM simulation data. In (Platt 2020) the author instead proposes the use of particle swarm samplers (Kaveh 2017; Stonedahl 2011), as well as the search heuristic of (Knysh and Korkolis 2016).

In this work, we take a different view of the problem and test the performance of existing search strategies, on a common calibration task, and propose simple methods to combine them in mixed strategies to drastically boost calibration performance. We test our methods one of the most well-known and studied macroeconomic ABMs (Delli Gatti et al. 2011a; Assenza, Delli Gatti, and Grazzini 2015; Dawid and Delli Gatti 2018), often referred to as the *CATS* ("Complex Adaptive Trivial System") model. Our contributions are as follows:

- We verify that the macroeconomic ABM considered can be efficiently calibrated to reproduce a variety of real time series.
- We find that methods based on random forest machine learning surrogates are particularly effective searchers in the highly non-convex and discretely-changing loss function induced by ABMs.
- We find that combining together different search methods almost always provides better overall performance, and propose this as a convenient heuristic to apply in the ABM calibration practice.
- We introduce a simple reinforcement-learning technique to automatically aggregate any number of search methods in a single mixed strategy, and demonstrate the superior performance of this approach with respect to naive aggregation strategies.

In Section 2 we overview the CATS model, in Section 3 we describe the calibration technique considered here and the search methods that we employ individually and in combination, in Section 4 we describe our benchmarking experiments and the results obtained, in Section 5 we describe the reinforcement learning scheme we proposed to automatically aggregate existing methods, and demonstrate its performance, in Section 6 we verify that the calibrated model approximately reproduces the target real data, and in Section 7 we conclude.

In the interest of reproducibility, the code and the data used to generate the key results of this work are available to download as supplementary material of the paper.

2 Model description

The CATS model (Delli Gatti et al. 2011b; Assenza, Delli Gatti, and Grazzini 2015) consists of four classes of interacting agents: households, final-goods producing firms (C-firms), capital producing firms (K-firms) and banks. Figure 1a illustrates these classes of agents and the main mechanisms of interactions among them.

2.1 Household

The household sector consists of workers and capitalists. Each worker supplies one unit of labour inelastically. An unemployed worker randomly selects Z_e firms and takes the job at the firm with a vacant position on a first come first serve basis. Each worker receives wage w until laid off. Firms are owned by capitalists and they receive dividends and hold equity at those firms but do not work. When a firm becomes bankrupt, it is replaced by a new entrant firm and a capitalist provides equity. All of the households consume final goods and therefore participate in search and matching in the consumption market. They determine their consumption budget according to

$$C_{c,t} = \overline{Y}_{c,t} + \chi D_{c,t}, \tag{1}$$

where $\overline{Y}_{c,t}$ is the *permanent income* of the consumer c at time t, $D_{c,t}$ is the financial wealth deposited at a bank and $\chi \in (0,1)$ is the fraction of the bank deposit used for consumption. Unlike the standard macroeconomic models, *permanent income* is the weighted average of current and past incomes with exponentially decaying weights and follows

$$\overline{Y}_{c,t} = \xi \overline{Y}_{c,t-1} + (1-\xi) Y_{c,t} \tag{2}$$

where $Y_{c,t}$ is the actual income of period t and $\xi \in (0,1)$ is the *memory parameter* of the consumer.
Each consumer visits a set of randomly selected firms and sorts their prices from lowest to highest (this gives rise to implicit negative relative price elasticity of demand). If the consumption budget is not exhausted on the first firm, the consumer goes to the second firm in the order. If consumption budget is not exhausted after all buying opportunities, the consumer involuntarily saves the rest.

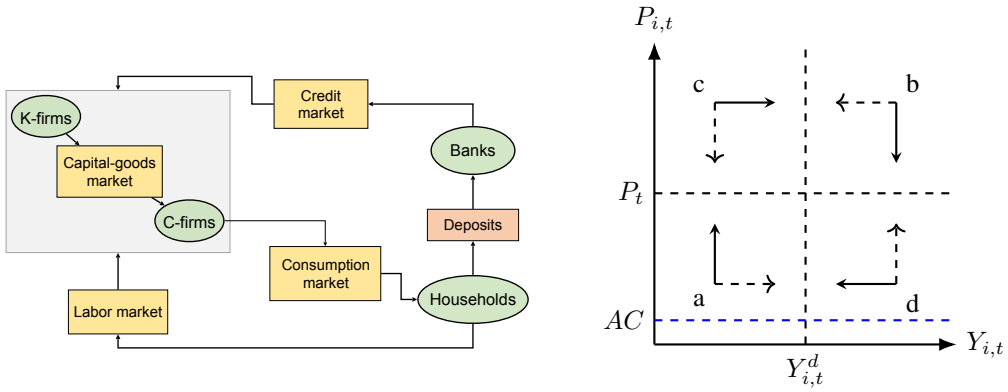
2.2 Price and quantity setting

One of the distinctive features of the CATS model is its expectation formation of the future demand and price setting of the firms, summarised in Figure 1b and detailed in this section. C-firms and K-firms decide the quantity and price in a similar fashion. The only difference between these two is that C-goods are non-storable, unlike K-goods. Firms start off with the pair $(P_{i,t}, Y_{i,t})$ and notice the actual sale $Q_{i,t} = \min(Y_{i,t}, Y^d_{i,t})$ as demanded quantity can differ from produced quantity. Therefore, firms base their decision on two signals: their relative price and actual sale. Now any decision can be mapped to one of the quadrants of the $(P_{i,t}, Y_{i,t})$ space depending on the signal. Hence firm $i \in \{\text{C-firms}, \text{K-firms}\}$ update their next period desired output as

$$Y^*_{i,t+1} = \begin{cases} Y_{i,t} + \rho(-\Delta_{i,t}) - \mathbb{1}_{i\in K} Y^k_{i,t+1} & \text{if } \Delta_{i,t} \le 0 \;\; P_{i,t} \ge P_t \text{ ('c')} \\ Y_{i,t} - \rho\Delta_{i,t} \mathbb{1}_{i\in K} Y^k_{i,t+1} & \text{if } \Delta_{i,t} > 0 \;\; P_{i,t} < P_t \text{ ('d')} \end{cases} \tag{3}$$

where $\Delta_{i,t} = Y_{i,t} - Y^d_{i,t}$, $\rho \in (0,1)$ and $Y^k_{i,t+1} = (1-\delta^k)(Y^k_{i,t} + \Delta_{i,t})$ i.e., the inventory dynamics of capital firms. Here $\delta^k \in (0,1)$ is the depreciation parameter of the inventories.

In short, when demand is higher than the current period's production, increase the next period's production and vice-versa. Notice that in the four possible signal scenarios depicted in the four quadrants of Figure 1b, firms can only change either prices or adjust their quantities. Equation (3) describes quadrants 'c' and 'd' of the figure, for the price



(a) (b)

Figure 1: The CATS model. (a) An illustration of the agent classes of the model and their interactions. Agent classes are represented in green ovals, interaction types are specified in rectangles, and markets are specified in yellow rectangles. The directions of the arrows indicate the flow of the specific good e.g., consumption-goods are acquired by households from C-firms, while labour is acquired by firms from households. (b) An illustration of the firms' decisions on the price-quantity space. Prices $P_{i,t}$ and quantities $Y_{i,t}$ of goods are updated following the 4 solid black arrows (representing Equations (4) and (3)), and not the dashed black arrows. The dashed blue line is the minimum price they can charge, corresponding to the average cost (AC) for production.

setting in the other two scenarios (quadrants 'a' and 'b' of the figure) firms follow the updating rule

$$P_{i,t+1} = \begin{cases} P_{i,t}(1+\eta_{i,t+1}) & \text{if } \Delta_{i,t} \leq 0 \;\; P_{i,t} < P_t \text{ ('a')} \\ P_{i,t}(1-\eta_{i,t+1}) & \text{if } \Delta_{i,t} > 0 \;\; P_{i,t} \geq P_t \text{ ('b')} \end{cases} \tag{4}$$

where $\eta_{i,t+1} \sim \mathcal{U}(0, \bar{\eta})$. So when there is excess demand, firms increase their price if it is lower than average, since consumers will be willing to pay a higher price and vice-versa. Firms also have average costs (AC) and can not set the price below the level of AC. C-firms produce taking the output of the K-firms as input and therefore participate in the K-goods market using search and match exactly like in consumption goods market.

2.3 Production, investment and employment

Means of production in the C-firms are capital $K_{i,t}$ and labour $N_{i,t}$. The production function follows Leontief technology i.e., $\hat{Y}_{i,t} = \min\,(\alpha N_{i,t}, \kappa K_{i,t})$ where α and κ are labor and capital productivity respectively. If the labour is abundant and capital is not fully utilized then the output becomes $Y_{i,t} = \omega_{i,t}\hat{Y}_{i,t} = \omega_{i,t}\kappa K_{i,t}$ where $\omega_{i,t} \in (0,1)$ is the *capacity utilization rate*. Therefore the required labor for the production is $N_{i,t} = (\kappa/\alpha)\omega_{i,t}K_{i,t}$. Capital is accumulated by the firms and follows

$$K_{i,t+1} = (1 - \delta\omega_{i,t})K_{i,t} + I_{i,t} \tag{5}$$

where only utilized capital depreciates and $I_{i,t}$ is the investment.

Investment opportunities of the firms are infrequent (one in every $1/\gamma$ periods where γ is the fraction of firms adjusting capital) and capital is fixed in the short run. This gives rise to sticky and durable capital, as firms take investment decisions in an uncertain environment before the consumption market opens and this anchors decisions on average lifetime capital stock. The average lifetime capital stock evolves as

$$\overline{K}_{i,t-1} = \nu\overline{K}_{i,t-2} + (1-\nu)\omega_{i,t-1}K_{i,t-1} \tag{6}$$

where $\nu \in (0,1)$.

Firms decide on investment in two parts. Firstly, they make up for the worn-out capital keeping in the mind the future opportunities of capital adjustment i.e $I^r_{i,t} = \frac{\delta}{\gamma}\overline{K}_{i,t-1}$. Secondly, they target the *desired long-term rate of capital utilization* $\overline{\omega}$. Therefore, the total investment of the firm becomes

$$I_{i,t} = \left(\frac{1}{\overline{\omega}} + \frac{\delta}{\gamma}\right)\overline{K}_{i,t-1} - K_{i,t} \tag{7}$$

and the capital stock evolves as:

$$K_{i,t+1} = \left(\frac{1}{\overline{\omega}} + \frac{\delta}{\gamma}\right)\overline{K}_{i,t-1} - \delta\omega_{i,t}K_{i,t} \tag{8}$$

If the required capital for the desired level of production is lower than the available capital stock, the firm uses a fraction of the stock. If the required capital is higher than the available capital stock, the firm fully utilizes the stock but the level of production is not reached. Following these rules, we get the required number of workers as

$$N^*_{i,t+1} = \min\,\left(\frac{\kappa}{\alpha}K^*_{i,t+1}, \frac{\kappa}{\alpha}K_{i,t+1}\right), \tag{9}$$

where $K^*_{i,t+1}$ is the required capital for desired production level and $K_{i,t+1}$ is the available capital stock.

After deciding on the required number of workers to match the desired level of production, firms post vacancies as follows

$$\nu_{i,t+1} = \max\left(N^*_{i,t+1} - N_{i,t}, 0\right). \tag{10}$$

K-firms produce only using labour input from the workers and use linear technology $Y_{j,t} = \alpha N_{j,t}$. Hence labour requirement of the firm is $N_{j,t}/\alpha$. To make up for the required workers, firms post vacancies and compete with C-firms in the labour market for hiring.

2.4 Credits and banks

Each firm takes loans from the bank to fund its production when internal funding is in short supply. For C-firms there are typically two costs, the wage of the workers and the funding for investment whereas K-firms only acquire the cost of wage. Hence the required loans by the firms are

$$F_{i,t} = \max\ (wN_{i,t} - \mathbb{1}_{i \in C-firms} P_{k,t-1} I_{i,t} D_{i,t-1}, 0) \tag{11}$$

There is only one bank in the economy. It accepts all deposits from agents and does not provide deposit interests. Bank evaluates the financial soundness of the firms using the entire past data of the firm's balance sheet. For each firm f, it computes the following leverage ratio

$$\lambda_{f,t} = \frac{L_{f,t-1} + F_{f,t}}{E_{f,t-1} + L_{f,t-1} + F_{f,t}}. \tag{12}$$

The bank then estimates a logistic regression of the individual bankruptcy probability ϕ_f for each firm as $\phi_f = f(\lambda_f)$. Considering that the firms are paying θ fraction of their loan back each period, the bank sets the interest rates of loan for each bank as

$$r_{f,t} = \mu \left\{ \frac{1 + \frac{r}{\theta}}{\Phi(\theta, T_{f,t})} - \theta \right\}, \tag{13}$$

where $T_{f,t} = 1/\phi_{f,t}$ i.e., number of periods after which firm defaults. Optimization of the lending is done by considering a maximum admissible loss for the bank as a fraction of the bank's equity. If $\Delta L_{f,t}$ is the new extended loan to the firm then it follows that

$$\phi_f(\Delta L_{f,t} + L_{f,t-1}) \leq \zeta E^b_t, \tag{14}$$

and the maximum admissible loan for a firm f becomes

$$\overline{F}_{f,t} = \frac{\zeta E^b_t - \phi_f L_{f,t-1}}{\phi_f}. \tag{15}$$

In summary, if the loan requirement of the firm is less than the maximum admissible loan for that firm, the firm gets the full funding. On the other hand, if the loan requirement is higher than the maximum admissible loan, the bank lends only up to the limit and the firm has to cut down its hiring, production etc.

3 Calibration description

The calibration method we consider is composed of three main steps. First, a search method (from now on also called a *sampler*) suggests a set of parameters to explore, then a number of simulations are performed for each selected parameter, and finally a loss function is evaluated to measure the goodness of fit of the simulations with respect to the real time series. Iterating these three steps allows finding parameters corresponding to progressively lower loss values, and the parameter corresponding to the lowest loss value can be considered optimal.

We follow the *method of moments* paradigm (Franke 2009; Chen and Lux 2018) and use the following loss function (often called *distance* in the ABM literature) for all calibrations. This takes the form

$$L = \frac{1}{D} \sum_{d=1}^{D} \mathbf{g}_d^{\mathrm{T}} \mathbf{W}_d \mathbf{g}_d, \tag{16}$$

where $\mathbf{g}_d$ is the vector of difference between the real and the simulated moments of the one-dimensional time series d, and D is the total number of dimensions in the multi-dimensional time series considered. Different choices for the weighting matrices $\mathbf{W}_d$ have been proposed in the literature (Franke 2009; Franke and Westerhoff 2012). In this work we take the $\mathbf{W}_d$ matrices to be diagonal matrices with elements $(\mathbf{W}_d)_{ii}$ inversely proportional to the square of the real i-th moment of the one-dimensional time series d. This choice guarantees that the same weight is given to all moments considered, independently of the different scales or units of measure that the different moments might have. In essence, the loss function written in this way provides an estimate of the relative squared error between real and simulated moments.

Since we use a common loss function for all calibrations, the only difference between the calibration runs considered here is the choice of search method. We consider the following five search methods, all of which are implemented in *Black-it* (Benedetti et al. 2022), an open source library for ABM calibration [1]

Halton sampler (H)**.** This sampler suggests points according to the Halton series (Halton 1964; Kocis and Whiten 1997). The Halton series is a low-discrepancy series providing a quasi-random sampling that guarantees an evenly distributed coverage of the parameter space. As the method is very similar to a purely random search, we use it as a baseline for the more advanced search strategies analysed.

Random forest sampler (RF)**.** This sampler is a type of machine learning surrogate sampler. It interpolates the previously computed loss values using a random forest classifier (Bajer, Pitra, and Holeňa 2015), and it then proposes parameters in the vicinity of the lowest values of the interpolated loss surface. We use a random forest classifier with 500 independent estimators ("trees") and use 10 classes chosen as the 10 quantiles of the distribution of evaluated losses.

XG-boost sampler (XB)**.** This sampler is a machine learning surrogate sampler that interpolates loss values using an XG-Boost regression (Chen and Guestrin 2016), as proposed in (Lamperti, Roventini, and Sani 2018). We use a learning rate of 0.1, a maximum tree depth of 5, and 10 estimators.

Gaussian process sampler (GP)**.** This sampler is a machine learning surrogate sampler that interpolates loss values using a Gaussian process regression (Conti and O'Hagan

[1] https://github.com/bancaditalia/black-it

Param.	Description	Range
ξ	Memory parameter in consumption	0.5-1
χ	Wealth parameter in consumption	0-0.5
ρ	Quantity adjustment	0-1
$\bar{\eta}$	Price adjustment	0-1
μ	Bank's gross mark-up	1-1.5
ϕ	Bank's leverage	0-0.01
δ^k	Inventories depreciation rate	0-0.5
γ	Fraction of investing C-firms	0-0.5
θ	Rate of debt reimbursement	0-0.1
ν	Memory parameter in investment	0-1
t_w	Tax rate	0-0.4

Table 1: Parameter descriptions and their corresponding calibration ranges.

2010; Rasmussen 2004). We use a Matérn covariance function with $\nu = 5/2$ and with the lengthscale optimised at every iteration via maximum marginal likelihood.

Best batch sampler (BB)**.** This sampler is a very essential type of genetic algorithm (Stonedahl 2011) that takes the parameters corresponding to the current lowest loss values and perturbs them slightly in a purely random fashion to suggest new parameter values to explore. The random perturbation is specifically obtained by first selecting a random subset of dimensions, and then changing the parameter value along those dimensions uniformly but within a short range (plus/minus 0.006 in our case).

4 Benchmarking experiments

4.1 Experiments preparation

Similarly to (Delli Gatti and Grazzini 2020), we calibrate the model using the following 5 historical time series, representing the US economy from 1948 to 2019, downloaded from the FRED database (McCracken and Ng 2016): total output, personal consumption, gross private investment (all in real terms), the implicit price deflator and the civilian unemployment. To make simulated and observed data comparable, we remove the trend component from the total output, consumption and investment using an HP filter (Ravn and Uhlig 2002); and we use simulated and observed price deflator to compute de-meaned inflation rates.

In Table 1 we list the 11 parameters considered for calibration and the specified ranges of variation.

4.2 Experiments performed.

Using the four samplers described in the previous section, we build 11 search methods as the 5 samplers taken individually, as well as the 6 combinations resulting from taking two different non-baseline samplers.

For each search method, we perform 3 independent calibration runs. Each calibration run consists of 3600 model evaluations, and for each parameter 5 independent simulations are performed to reduce the statistical variance of the loss estimate. Each simulated series consists of 800 timesteps generated by running the model for 1100 time steps and discarding the first 300. This makes up a total of 540000 simulations and more than 50 days of CPU time, which we were able to compress in less than two days by leveraging parallel computing both within and between calibrations.

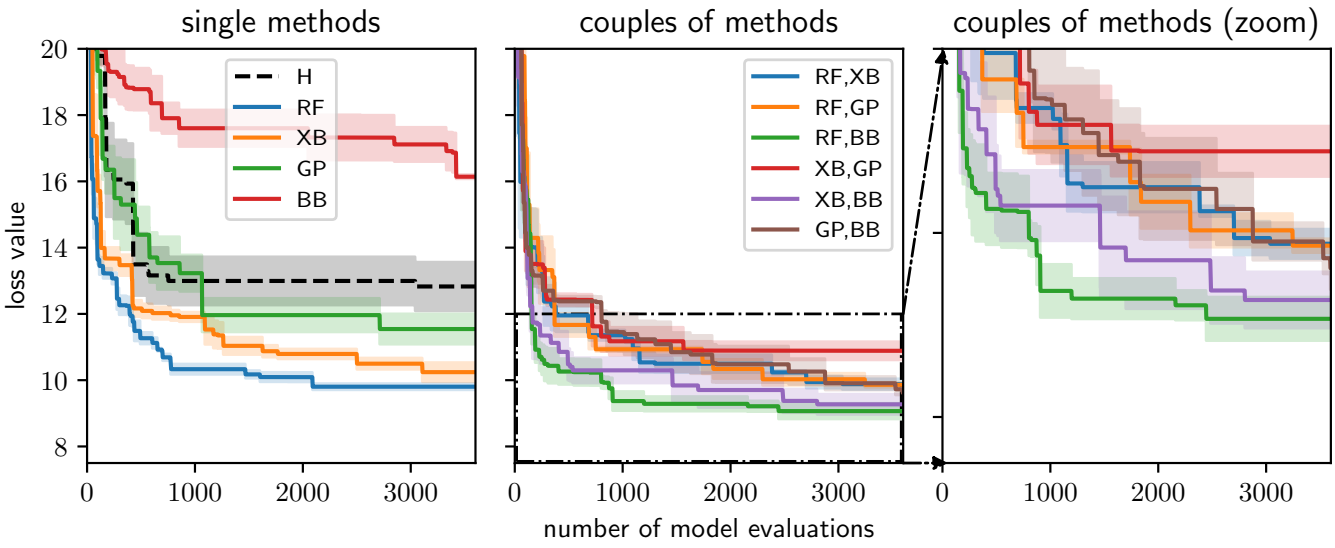
4.3 Results and discussion

Figure 2 reports the cumulative minimum loss achieved by the different sampling strategies as a function of the number of model evaluations performed. The lines and the shaded areas indicate averages and standard errors over the 3 realisations of the experiment. Single samplers are reported in the left graph, while couples of samplers are reported in the middle graph as well as –zoomed– in the right graph. The table at the bottom of the graphs reports the minimum loss achieved by the different methods.

Single methods. When samplers are taken in isolation, the random forest sampler (RF) clearly outperforms all other methods, the XG-boost sampler (XB) is the second best performing and the Gaussian process sampler (GP) is substantially worse than the other two machine learning surrogate samplers. The low performance of the GP sampler can be ascribed to the smoothness and regularity assumptions inherent in Gaussian process regression models, assumptions that are not present in random-forest or XG-boost models, and not suited to describe the roughed and complex loss landscape of ABM calibrations. The best batch sampler (BB) performs very poorly in isolation, and underperforms even in comparison with the baseline H sampler. This is not entirely surprising, since the BB sampler can only propose small perturbations around current loss minima and can thus easily remain stuck in one of the many local minima of the highly non-convex landscape typical of ABMs loss functions.

Couples of methods. All methods, not just the poorly performing BB sampler, possess intrinsic sampling biases that in the long run can hinder their performances and make them converge to sub-optimal solutions. We find that combining different methods in mixed strategies can strongly mitigate such biases and improve overall performance. The effect can be observed in the second and third panel of Figure 2, by noticing that couples of methods, with the only exception of the 'XB, GP' combination, always perform on par or better than the best single samplers (RF and XB). Interestingly, the best overall performances are achieved by coupling one machine learning surrogate sampler with the genetic BB sampler. In light of the above discussion, we note that machine learning surrogate samplers and the BB sampler work in very different ways, and hence their combination can strongly diminish the respective sampling biases, while since machine learning surrogates all work in similar ways, their combination does not yield to comparable improvements. The RF, BB and the XB, BB combinations are particularly effective and achieve the lowest loss values.

To summarise, our results show that the RF and XB samplers are particularly well suited to efficiently search in the parameter space of ABMs. The success of the RF and XB samplers can be ascribed to the ability to correctly approximate high dimensional and possibly discontinuous functions



Method	H	RF	XB	GP	BB	RF,XB	RF,GP	RF,BB	XB,GP	XB,BB	GP,BB
Mean	12.83	**9.803**	10.24	11.96	16.87	9.88	9.861	**9.07**	10.89	9.27	9.83
Std. Err.	0.73	0.094	0.29	0.51	0.59	0.16	0.075	0.24	0.27	0.31	0.26

Figure 2: Top graphs: Loss as a function of the number of model evaluations for the single methods (left), and for couples of methods (middle and right). Bottom table: Means and standard errors of the lowest losses achieved by the different strategies. Note that these results are directly comparable with those shown in Figure 3 and discussed in the next section, as both x and y axes have identical ranges.

with no regularities. However, the performance of the RF and XB samplers can be significantly improved if they are used in combination with the BB sampler.

The results presented so far can already offer useful guidance for researchers interested in calibrating medium and large scales ABMs, as they provide an easy recipe to boost calibration efficiency by simple alternation of existing search methods. In the next section, we move a step forward and consider the combination of multiple methods in more general terms, without limiting ourselves to the simplest scenario of a "round-robin" selection.

5 Reinforcement learning experiments

The results of the benchmarks presented in Section 4 show that the combination of different types of sampling methods can be beneficial for the calibration process even when we naively alternate the available sampling methods during the course of a calibration. This suggests that the investigation of different –and more flexible– scheduling policies of search methods could bring to even more efficient calibrations.

In particular, it is desirable that the chosen scheduling policy shows some form of *adaptivity*, i.e. that is able to choose the sampling method with more chances to sample a good parameter vector, taking into account the progress of the calibration process. To achieve this goal, we frame the ABM calibration problem as a reinforcement learning (RL) problem where the decision-maker (the *agent*) has to find a good policy such that it chooses the most promising search method, where "promising" is related to the chances of sampling a parameter that improves the value of the loss. The decision-maker receives feedback for its choice in the form of a reward signal computed from the sampled loss function values. This is what makes the scheduling policy adaptive: search methods that more often provide loss improvements are more rewarding from the decision-maker perspective, and they have more chances of being chosen in the next calibration step; on the other hand, whenever a search method does not show to be rewarding anymore, then the decision-maker can detect this and switch the preference to another search method. Borrowing terminology from control theory (Dorf and Bishop 2008), fixed scheduling policies, such as the naive samplers' combinations explored in the previous section, are *open-loop*, i.e. they do not change regardless of how a search method is performing; instead, RL-based scheduling policies are *closed-loop*, because they receive and process the feedback coming from the calibration process, possibly reacting to such feedback by changing the preferred sampling method.

Specifically, we frame the calibration process as a *multi-armed bandit (MAB)* problem (Katehakis and Veinott Jr 1987; Weber 1992; Auer, Cesa-Bianchi, and Fischer 2002; Berry and Fristedt 1985; Gittins, Glazebrook, and Weber 2011; Lattimore and Szepesvári 2020). This is a classic reinforcement learning problem that exemplifies the *exploration–exploitation trade-off dilemma* (Sutton and Barto

2018). The challenge for the agent is to simultaneously attempt to acquire new knowledge by "exploring" different *actions* and optimise their decisions based on existing knowledge by "exploiting" actions that have been estimated to be rewarding. We define the different sampling methods as the actions available for the agent, and loss improvements as the reward signals. More formally, we define the reward at time t as the fractional improvement achieved over the previous best loss

$$R_t = \max\{0, \frac{L_{\mathsf{best},t-1} - L_t}{L_{\mathsf{best},t-1}}\} \tag{17}$$

where L_t is the loss obtained for the simulations sampled at time t, and $L_{\mathsf{best},t-1}$ is the best loss sampled so far up to time $t-1$. Note that R_t is a random variable, because L_t depends on the simulated time series outputted by the (possibly stochastic) ABM, and the (possibly stochastically) chosen parameter vector. As in most of the MAB problems, the goal for the agent is to maximize the cumulative sum of rewards

$$S_N = \sum_{t=1}^{N} R_t, \tag{18}$$

where N is the number of calibration steps.

Differently from the usual MAB setting, the reward probability distributions associated to each available sampler are obviously *non-stationary*, and in fact they change drastically during the course of the calibration. As an example, consider that at end of a calibration all methods –even the best ones– stop providing any improvement in the loss, and hence the reward distributions become progressively more peaked around zero. Non-stationarity is the most general assumption one can make over the behaviour of reward probability distributions in MABs (Auer et al. 2002) and, in our case, the non-stationarity assumption is required from of the lack of knowledge on both the ABM and the samplers' behaviours.

The MAB is a very simple framework for RL problems, that are more generally modelled as Markov Decision Processes (MDPs) (Sutton and Barto 2018). However, their simplicity is precisely what makes MAB better suited for our context than other approaches. Indeed, as MAB algorithms focus on finding the best action at each step rather than learning the entire environment, they are much more sample efficient. In the ABM calibration context, simulations are typically very expensive, and consequently the sample efficiency of the learning method is of paramount importance.

In the following, we test our MAB framework in two experiments. First, in the *offline-learning* experiments, we let the agent learn from the previously executed calibrations of Section 4. Then, in the *online-learning* experiments, we let the agent interact with the environment and optimise its policy on-the-fly during each calibration.

5.1 Offline experiments

In this section, we train a MAB agent over past calibration histories. More precisely, we take the single methods and couples of methods calibrations of Section 4, and process them as if they were observed by a MAB algorithm. This approach gives us an estimate of the expected gain of each sampler, and therefore information about the effectiveness of the sampler methods on the specific calibration task.

Sampler \ Context	sing. samp.	glob.	high $L_{\mathsf{best},t}$	low $L_{\mathsf{best},t}$
RF	**0.25**	**0.27**	**1.3**	0.052
XB	0.23	0.23	0.61	0.033
GP	0.21	0.17	0.26	0.068
BB	0.11	0.23	0.28	**0.18**
H	0.20	0.20	0.24	N.A.

Table 2: The estimated Q functions for the different search methods and under different contexts. (sing. samp/) uses only single sample calibrations, (glob.) uses all calibrations, (high $L_{\mathsf{best},t}$) uses all calibration but only actions taken when the loss is *above* the median loss, and (low $L_{\mathsf{best},t}$) uses all calibration but only actions taken when the loss is *below* the median loss. Results are reported on a scale of 10^{-3}.

In the context of MAB solutions, *action-value methods* are methods for estimating the values of actions and for using the estimates to make action selection decisions (Sutton and Barto 2018). Let $Q(a)$ be the value of action a or, in our context, the value of using a specific search method during a calibration. One natural way to estimate such values is by averaging the rewards actually received

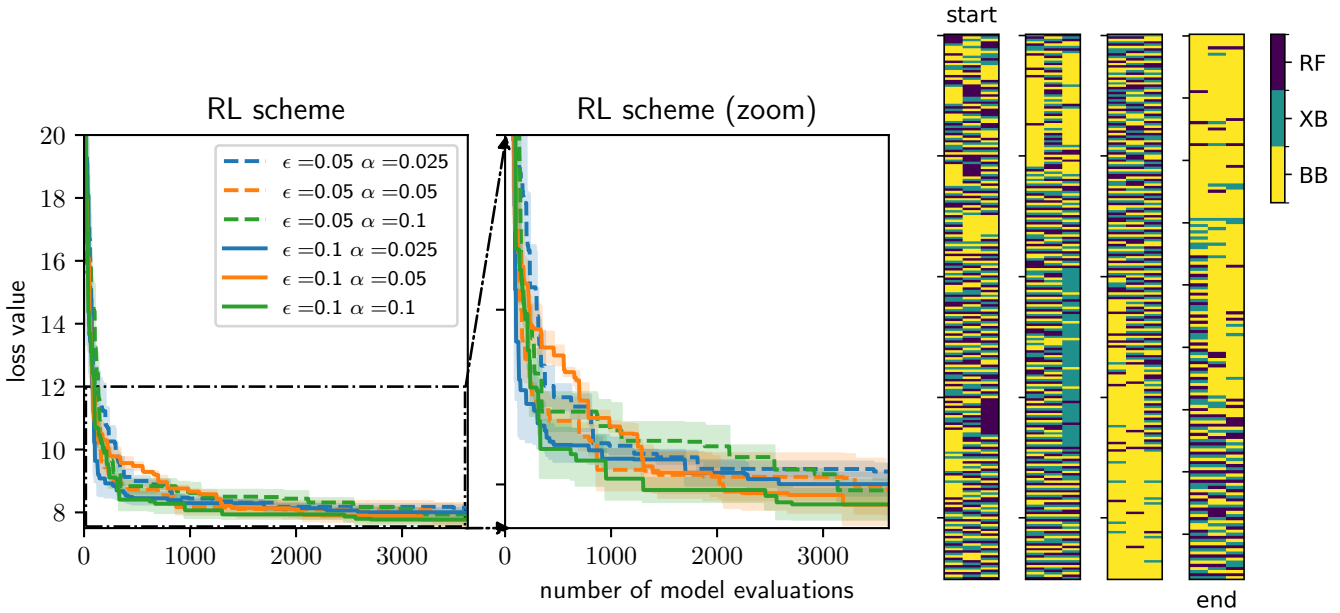
$$Q(a) = \frac{\sum_{t=1}^{N} R_t \cdot \mathbb{1}_{A_t=a}}{\sum_{t=1}^{N} \mathbb{1}_{A_t=a}}, \tag{19}$$

where A_t is the action chosen at step t. This approach is often called the *sample-average* method (Sutton and Barto 2018).

The first two columns of Table 2 provide the results of this analysis when only the single sampler calibrations are considered ("sing. samp." column) and when all calibrations are considered ("glob." column). Not surprisingly, the RF sampler reaches the highest Q value using both datasets, and the results of the "sing. samp." column replicate the hierarchy of samplers of the first panel of Figure 2. Interestingly, the value of the BB sampler dramatically increases when the combined dataset is used, confirming the analysis carried forward in the last section on the effectiveness of using the BB sampler in combination with a machine learning surrogate sampler.

The third and fourth columns of Table 2 offer additional insight. In these columns, we restrict the value function estimation of Eq. (19) to actions performed in one of two different 'states', characterised by the best loss $L_{\mathsf{best},t}$ being either above the median ("high $L_{\mathsf{best},t}$" column) or below the median ("low $L_{\mathsf{best},t}$" column). Models of this kind, where the actions of a MAB agent depend on one or more states (in this case high/low loss value) are known as *contextual* MABs (Langford and Zhang 2007; Lu, Pal, and Pal 2010; Li et al. 2010).

The results clearly indicate that when the loss is high (typically at the beginning of the calibration) the optimal action is the RF sampler, but when the loss is low (typically at the end of the calibration) the optimal action becomes, by far,



$\epsilon \backslash \alpha$	0.025	0.05	0.1
0.05	8.148 ± 0.089	7.94 ± 0.28	7.93 ± 0.12
0.1	8.00 ± 0.13	$\mathbf{7.77} \pm 0.23$	$\mathbf{7.77} \pm 0.17$

Figure 3: Top graphs: (left and middle) Loss as a function of the number of model evaluations for the RL scheme with different choices of parameters, (right) the specific actions (samplers) selected by the RL scheme with parameters $\epsilon = 0.1$ and $\alpha = 0.1$ during the 900 epochs of a calibration for each of the 3 independent runs, to be read from left to right, from top to bottom, note that each epoch provides 4 model evaluations. Bottom table: Means and standard errors of the lowest losses achieved by the RL scheme. These results can be compared directly with those of Figure 2 as they have identical ranges on both x and the y axes.

the BB sampler. The BB sampler proposes small perturbations around low-loss parameter combinations, and hence it can be expected to be particularly effective when the calibration has already reached a good minimum, which can be further explored with this method.

The analysis performed so far would suggest the design of a mixed search scheme that exploits a machine learning surrogate sampler (say RF or XB) when the loss is sufficiently high, before switching to the BB sampler towards the end of the calibration. However, this specific strategy would not be generally applicable as, on a new calibration task, one would not know in advance the loss values that can be achieved, and hence could not set any loss threshold on the choice of sampler. In the following section, we show how a MAB agent trained on-the-fly can solve this problem by learning this behaviour, without any prior information, during the course of a single calibration run.

5.2 Online experiments

In online learning schemes, the agent interacts with the environment through a specific policy π while simultaneously optimising the policy. We propose the use of one of the most well-known algorithms for online learning of MAB agents in non-stationary environments: the ϵ-greedy policy with fixed learning rate (Sutton and Barto 2018). In this framework, at each step t, with large probability $1 - \epsilon$ the agent performs a 'greedy' action i.e., it selects the action a with the highest value $Q(a)$, and with small probability ϵ it selects a purely random action. We can hence write down the ϵ-greedy MAB policy as follows

$$\pi_t = \begin{cases} \text{argmax}_a \; Q_t(a) & \text{with probability } 1 - \epsilon \\ \text{random action} & \text{with probability } \epsilon \end{cases}. \quad (20)$$

After the selected action a is performed, the agent receives a reward R_t, and updates the value $Q_t(a)$ as

$$Q_{t+1}(a) = \alpha R_t + (1 - \alpha) Q_t(a), \quad (21)$$

where α is referred to as the *learning rate*. Note that the above update rule can be seen as an exponentially weighted moving average of the rewards obtained through action a. The exponential weighting guarantees that the current value of the Q function is not substantially affected by rewards received many steps earlier and, in turn, this allows the algorithm to adapt to changes of the environment on-the-fly during a calibration.

Figure 3 shows the results obtained when using the described scheme with a set of possible actions given by the tree samplers RF, XB and BB. The left and middle panels of the figure can be directly compared with the graphs in Figure 2, as they have identical ranges on both x and y axes. We see that the RL scheme proposed strongly outperforms

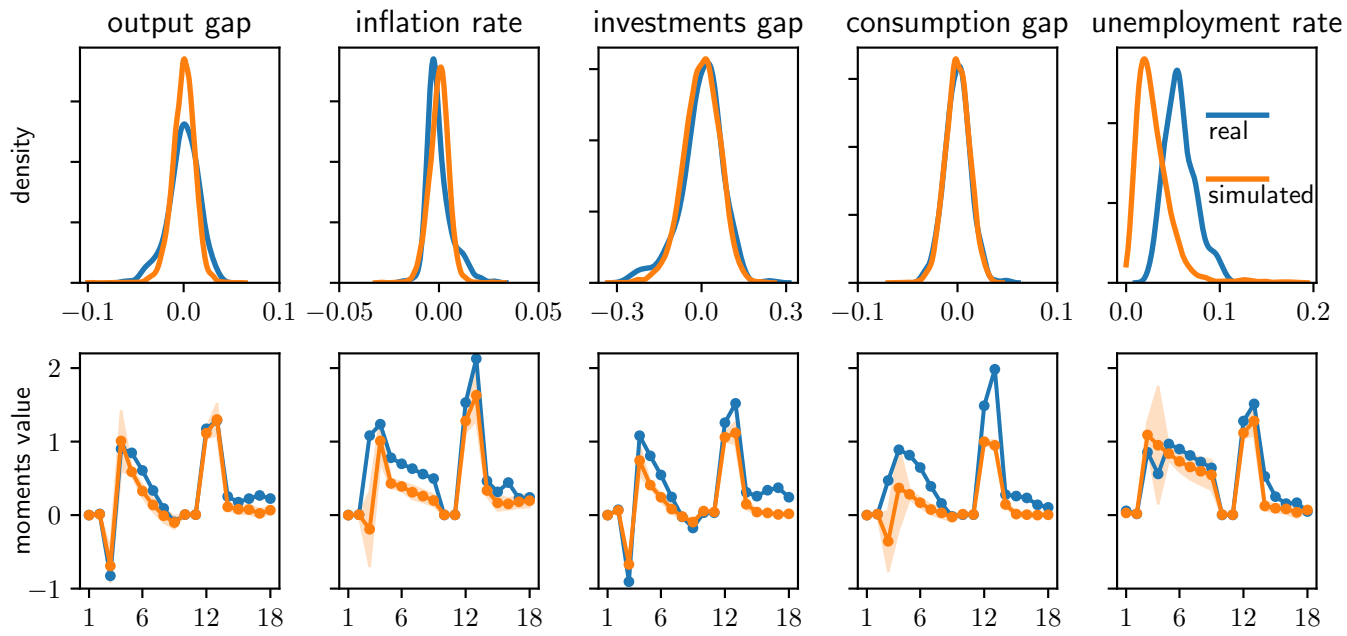


Figure 4: A comparison between distributions and moments of the real series (blue) and the simulated series of lowest loss (orange). The first row reports density estimates obtained via a kernel density estimator. The second row reports the value of the moments. In the second row, the indices from 1 to 18 on the x-axis represent the following statistics. 1-4): mean, variance, skewness and kurtosis, 5-9): autocorrelations of increasing time lags. 10-14): mean, variance, skewness and kurtosis of the differentiated time series, 14-18): autocorrelations of the differentiated time series.

any other method, or method combination, tested in the previous section. This happens for all values considered for the parameters ϵ and α, with the best results –by a very narrow margin– obtained with $\epsilon = \alpha = 0.1$.

The right panel of Figure 3 helps us build intuition around the excellent performance of the RL scheme proposed. It depicts with different colours the different actions (samplers) selected during the 3 RL calibration runs performed with the best parameters $\epsilon = \alpha = 0.1$. At the beginning of the calibration (say, the first two columns), the agent explores the different strategies by alternating between the 3 samplers and sometimes exploits a specific sampler with long streaks of identical sampler choices. Towards the end of the calibration (say, the last two columns), when the loss is low, the agent instead more decisively exploits the BB sampler, in agreement with the offline experiments described earlier and summarised in Table 2.

In conclusion, we find that modelling the calibration process as an online learning MAB problem, with actions being given by different available search methods, allows to detect the most promising search methods during the course of a single calibration. This gives rise to a very efficient sampling scheme, and represents a practical tool to intelligently combine different search methods in the calibration of economic ABMs.

The ϵ-greedy –fixed learning rate– scheme we use here is a particularly simple and intuitive algorithm for MAB learning, but many other options have been suggested in the literature. In Appendix A we explore some of them for a simplified calibration setting, and find no substantial improvements in the calibration efficiency.

6 Validation

We here verify that the calibrated model is able to approximately reproduce the behaviour of the five variables tracked in the real dataset. This can be immediately seen by analysing Figure 4, in which the distribution and the moments of the simulated series with the lowest loss are compared with those computed for the real historical series. In agreement with (Delli Gatti and Grazzini 2020), output, consumption and investment are very well captured by the CATS model, while stronger deviations can be observed in inflation and unemployment rates. Also in agreement with (Delli Gatti and Grazzini 2020) we find that, in general, the CATS model can only partially account for the persistence of the real time series. This is clear from the fact that the simulated series have systematically lower values of virtually all autocorrelations considered (indices 5-9 and 14-18 in the second-row graphs).

7 Conclusions

In this work, we systematically compare the performance of 5 search strategies, taken in isolation and in combination, on a method-of-moments calibration of a standard macroeconomic ABM. Our results show that calibration based on machine learning surrogate samplers, of the kind proposed

in (Lamperti, Roventini, and Sani 2018) but using a random forest algorithm for interpolation, provides superior performance with respect to the other search methods. Our results further show that coupling different search methods together gives rise to search strategies that typically improve over their constituents. The empirical efficacy of random forest search methods and of combining different search methods can be of practical help to researchers interested in calibrating and using medium and large-scale economic ABMs. However, when combining different search methods a natural issue arises about which methods should be combined, and in which way.

We provide a solution to this issue by framing the choice of search methods as a multi-armed bandit problem, and leveraging a well-known reinforcement learning scheme to select the best method on-the-fly during the course of a single calibration. The RL scheme proposed outperforms any other method or method combination tested, and thus provides a practical tool for researchers interested in efficiently calibrating ABMs.

In the future, it would be interesting to deepen the analyses of the present study in two possible lines of research, based on either extensions of the banchmarking experiments of Section 4 or on further investigations into the RL scheme of Section 5.

The benchmarking framework could be extended in several dimensions. The first is the testing of other standard search methods, such as particle swarm samplers or machine learning samplers based on neural networks. The second is the inclusion in the analysis of other measures of goodness of fit, in addition to the method of moments, such as likelihood measures, Bayesian measures (Grazzini, Richiardi, and Tsionas 2017; Farmer et al. 2022), or information theoretic measures (Lamperti 2018). The third is the addition of other widely known macroeconomic ABMs (Dawid and Delli Gatti 2018) to the analysis, such as the so called "K+S" model (Dosi, Fagiolo, and Roventini 2010), or the recent large-scale model of (Poledna et al. 2023). This would allow quantitative benchmarking not only of the calibration strategies, but also of the different models when calibrated on the same data. The final direction would involve appropriately increasing the data on which the ABMs are calibrated and tested, potentially with more variables and with more national economies. In essence, while the present work is an important step towards a systematic assessment calibration methods for medium and large-scale economic ABMs, all of the above mentioned directions would surely represent equally important steps towards an increasingly more data-driven ABM development.

Given the excellent results achieved, the RL scheme proposed also deserves further specific investigation. For example, one could verify whether the RL search method developed here maintains its high performance also in the more general setting of black-box function optimisation, perhaps in other specific application domains that might have peculiarities similar to the ABM calibration problem. One might also try to extend the simple (yet effective) MAB framework introduced here, by providing more 'contextual' information to the agent and hence attempting to represent the ABM calibration problem either as an online contextual-MAB problem, or directly as a partially-observable MDP (Kaelbling, Littman, and Cassandra 1998). Potentially, the problem could even be made suited for a pure MDP formulation by feeding the entire history of the past sampled point to the agent that needs to decide on the next search method, or directly decide the specific points to sample as proposed in (Chen et al. 2017).

References

Allesiardo, R.; and Féraud, R. 2015. Exp3 with drift detection for the switching bandit problem. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 1–7. IEEE.

Angione, C.; Silverman, E.; and Yaneske, E. 2022. Using machine learning as a surrogate model for agent-based simulations. *Plos one*, 17(2): e0263150.

Assenza, T.; Delli Gatti, D.; and Grazzini, J. 2015. Emergent dynamics of a macroeconomic agent based model with capital and credit. *Journal of Economic Dynamics and Control*, 50: 5–28. Crises and Complexity.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2): 235–256.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The Nonstochastic Multiarmed Bandit Problem. *SIAM J. Comput.*, 32(1): 48–77.

Axtell, R. L.; and Farmer, J. D. 2022. Agent-based modeling in economics and finance: Past, present, and future. *Journal of Economic Literature*.

Bajer, L.; Pitra, Z.; and Holeňa, M. 2015. Benchmarking Gaussian processes and random forests surrogate models on the BBOB noiseless testbed. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 1143–1150.

Baptista, R.; Farmer, J. D.; Hinterschweiger, M.; Low, K.; Tang, D.; and Uluc, A. 2016. Macroprudential policy in an agent-based model of the UK housing market.

Benedetti, M.; Catapano, G.; De Sclavis, F.; Favorito, M.; Glielmo, A.; Magnanimi, D.; and Muci, A. 2022. Black-it: A Ready-to-Use and Easy-to-Extend Calibration Kit for Agent-based Models. *Journal of Open Source Software*, 7(79): 4622.

Berry, D. A.; and Fristedt, B. 1985. Bandit problems: sequential allocation of experiments (Monographs on statistics and applied probability). *London: Chapman and Hall*, 5(71-87): 7–7.

Besson, L. 2018. SMPyBandits: an Open-Source Research Framework for Single and Multi-Players Multi-Arms Bandits (MAB) Algorithms in Python. Online at: github.com/SMPyBandits/SMPyBandits. Code at https://github.com/SMPyBandits/SMPyBandits/, documentation at https://smpybandits.github.io/.

Bookstaber, R.; Paddrik, M.; and Tivnan, B. 2014. An agent-based model for financial vulnerability. Technical report, Office of Financial Research Working Paper Series.

Bubeck, S.; Cesa-Bianchi, N.; et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1): 1–122.

Carro, A. 2022. Could Spain be less different? Exploring the effects of macroprudential policy on the house price cycle.

Catapano, G.; Franceschi, F.; Loberto, M.; and Michelangeli, V. 2021. Macroprudential policy analysis via an agent based model of the real estate sector. *Bank of Italy Temi di Discussione (Working Paper) No*, 1338.

Chan-Lau, M. J. A. 2017. *ABBA: An agent-based model of the banking system*. International Monetary Fund.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Chen, Y.; Hoffman, M. W.; Colmenarejo, S. G.; Denil, M.; Lillicrap, T. P.; Botvinick, M.; and Freitas, N. 2017. Learning to learn without gradient descent by gradient descent. In *International Conference on Machine Learning*, 748–756. PMLR.

Chen, Z.; and Lux, T. 2018. Estimation of sentiment effects in financial markets: A simulated method of moments approach. *Computational Economics*, 52(3): 711–744.

Cokayne, G. 2019. The effects of macroprudential policies on house price cycles in an agent-based model of the Danish housing market. Technical report, Danmarks Nationalbank Working Papers.

Conti, S.; and O'Hagan, A. 2010. Bayesian emulation of complex multi-output and dynamic computer models. *Journal of statistical planning and inference*, 140(3): 640–651.

Covi, G.; Montagna, M.; and Torri, G. 2020. On the Origins of Systemic Risk. Technical report, European Central Bank Working Papers.

Dawid, H.; and Delli Gatti, D. 2018. Agent-based macroeconomics. *Handbook of computational economics*, 4: 63–156.

Delli Gatti, D.; Desiderio, S.; Gaffeo, E.; Cirillo, P.; and Gallegati, M. 2011a. *Macroeconomics from the Bottom-up*, volume 1. Springer Science & Business Media.

Delli Gatti, D.; Desiderio, S.; Gaffeo, E.; Cirillo, P.; and Gallegati, M. 2011b. *Macroeconomics from the Bottom-up*, volume 1. Springer Science & Business Media.

Delli Gatti, D.; and Grazzini, J. 2020. Rising to the challenge: Bayesian estimation and forecasting techniques for macroeconomic Agent Based Models. *Journal of Economic Behavior & Organization*, 178: 875–902.

Dorf, R. C.; and Bishop, R. H. 2008. *Modern control systems*. Pearson Prentice Hall.

Dosi, G.; Fagiolo, G.; and Roventini, A. 2010. Schumpeter meeting Keynes: A policy-friendly model of endogenous growth and business cycles. *Journal of Economic Dynamics and Control*, 34(9): 1748–1767.

Fagiolo, G.; Moneta, A.; and Windrum, P. 2007. A critical guide to empirical validation of agent-based models in economics: Methodologies, procedures, and open problems. *Computational Economics*, 30(3): 195–226.

Farmer, J. D.; Dyer, J.; Cannon, P.; Schmon, S.; et al. 2022. Black-box Bayesian inference for economic agent-based models. Technical report, Institute for New Economic Thinking at the Oxford Martin School, University ....

Franke, R. 2009. Applying the method of simulated moments to estimate a small agent-based asset pricing model. *Journal of Empirical Finance*, 16(5): 804–815.

Franke, R.; and Westerhoff, F. 2012. Structural stochastic volatility in asset pricing dynamics: Estimation and model contest. *Journal of Economic Dynamics and Control*, 36(8): 1193–1211.

Garivier, A.; and Cappé, O. 2011. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, 359–376. JMLR Workshop and Conference Proceedings.

Gilli, M.; and Winker, P. 2003. A global optimization heuristic for estimating agent based models. *Computational Statistics & Data Analysis*, 42(3): 299–312.

Gittins, J.; Glazebrook, K.; and Weber, R. 2011. *Multi-armed bandit allocation indices*. John Wiley & Sons.

Grazzini, J.; and Richiardi, M. 2015. Estimation of ergodic agent-based models by simulated minimum distance. *Journal of Economic Dynamics and Control*, 51: 148–165.

Grazzini, J.; Richiardi, M. G.; and Tsionas, M. 2017. Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control*, 77: 26–47.

Halton, J. H. 1964. Algorithm 247: Radical-inverse quasi-random point sequence. *Communications of the ACM*, 7(12): 701–702.

Hommes, C.; He, M.; Poledna, S.; Siqueira, M.; and Zhang, Y. 2022. CANVAS: A Canadian Behavioral Agent-Based Model. Technical report, Bank of Canada.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2): 99–134.

Katehakis, M. N.; and Veinott Jr, A. F. 1987. The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research*, 12(2): 262–268.

Kaveh, A. 2017. Particle swarm optimization. In *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, 11–43. Springer.

Knysh, P.; and Korkolis, Y. 2016. Blackbox: A procedure for parallel optimization of expensive black-box functions. *arXiv preprint arXiv:1605.00998*.

Kocis, L.; and Whiten, W. J. 1997. Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software (TOMS)*, 23(2): 266–294.

Lamperti, F. 2018. An information theoretic criterion for empirical validation of simulation models. *Econometrics and Statistics*, 5: 83–106.

Lamperti, F.; Roventini, A.; and Sani, A. 2018. Agent-based model calibration using machine learning surrogates. *Journal of Economic Dynamics and Control*, 90: 366–389.

Langford, J.; and Zhang, T. 2007. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in neural information processing systems*, 20(1): 96–1.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A Contextual-Bandit Approach to Personalized News Article Recommendation. In *Proceedings of the 19th international conference on World wide web - WWW '10*, 661. ArXiv:1003.0146 [cs].

Lu, T.; Pal, D.; and Pal, M. 2010. Contextual Multi-Armed Bandits.

McCracken, M. W.; and Ng, S. 2016. FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics*, 34(4): 574–589.

Méro, B.; Borsos, A.; Hosszú, Z.; Oláh, Z.; and Vágó, N. 2022. A high resolution agent-based model of the hungarian housing market. *MNB Working Papers*, 7.

Plassard, R.; et al. 2020. Making a Breach: The Incorporation of Agent-Based Models into the Bank of England's Toolkit. Technical report, Groupe de REcherche en Droit, Economie, Gestion (GREDEG CNRS), Université ....

Platt, D. 2020. A comparison of economic agent-based model calibration methods. *Journal of Economic Dynamics and Control*, 113: 103859.

Platt, D. 2021. Bayesian estimation of economic simulation models using neural networks. *Computational Economics*, 1–52.

Poledna, S.; Miess, M. G.; Hommes, C.; and Rabitsch, K. 2023. Economic forecasting with an agent-based model. *European Economic Review*, 151: 104306.

Raj, V.; and Kalyani, S. 2017. Taming non-stationary bandits: A Bayesian approach. *arXiv preprint arXiv:1707.09727*.

Rasmussen, C. E. 2004. *Gaussian processes in machine learning*. Springer.

Ravn, M. O.; and Uhlig, H. 2002. On adjusting the Hodrick-Prescott filter for the frequency of observations. *Review of economics and statistics*, 84(2): 371–376.

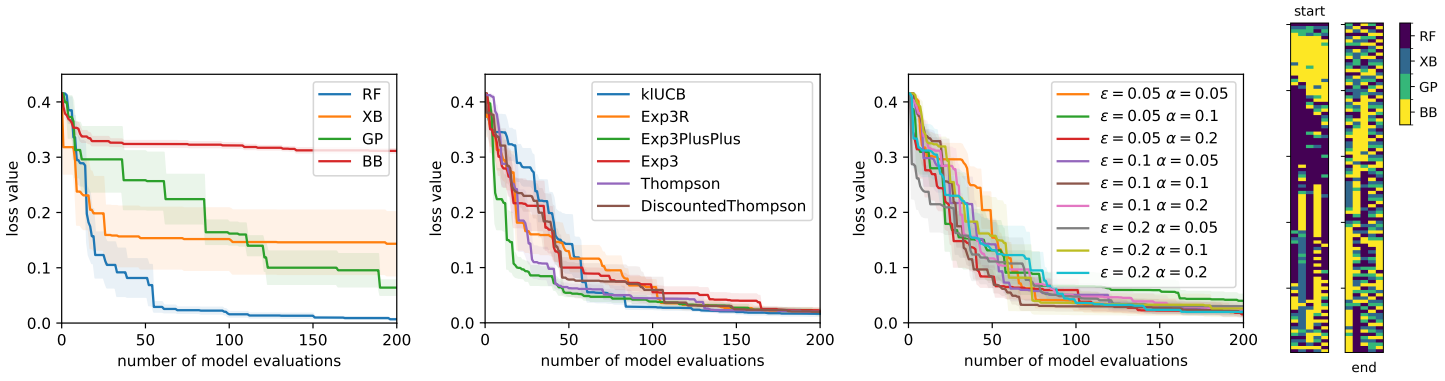
Seldin, Y.; and Slivkins, A. 2014. One practical algorithm for both stochastic and adversarial bandits. In *International Conference on Machine Learning*, 1287–1295. PMLR.

Stonedahl, F. J. 2011. *Genetic algorithms for the exploration of parameter spaces in agent-based models*. Ph.D. thesis, Northwestern University.

Sutton, R. S.; and Barto, A. G. 2018. *Reinforcement learning: An introduction*. MIT press.

Turrell, A. 2016. Agent-based models: understanding the economy from the bottom up. *Bank of England Quarterly Bulletin*, Q4.

Weber, R. 1992. On the Gittins index for multiarmed bandits. *The Annals of Applied Probability*, 1024–1033.



(a) (b) (c) (d)

Figure 5: Tests of the RL calibration framework with different MAB learning algorithms. (a)-(c) Loss values as a function of the number of model evaluations for different sampling strategies. Lines and shaded areas represent means and standard errors over 5 repetitions of each calibration run. (a) Baseline calibrations using the 4 different samplers individually. (b) RL calibrations using 6 different MAB learning algorithms. (c) RL calibrations using the fixed-α, ϵ-greedy learning algorithm proposed in the main text. (d) The specific 'actions' selected by the fixed-α, ϵ-greedy RL scheme with $\alpha = \epsilon = 0.1$ in the 5 calibration repetitions.

A A comparison of multiple MAB learning algorithms

In this appendix, we test the performance of a number of variations of the RL framework introduced in the main text obtained by coupling it with different learning algorithms for multi-armed bandits (MABs). For reasons of computational cost, the comparison is performed in a simplified setting, and not on the calibration of the economic ABM analysed in the rest of this work. The experimental setting consists of a method of moments calibration of a 5-state Markov process defined by a diagonal transition matrix, with 5 free parameters to calibrate. The target time series is generated by simulating the model for 5000 steps with diagonal transition parameters $(0.1, 0.2, 0.3, 0.4, 0.5)$. We define the action space of the MAB as the set of the 4 samplers (RF,XB,GP,BB).

Figure 5a shows the baseline calibrations obtained using the 4 samplers individually, and we see that also for this model the RF sampler outperforms all other search methods. Figure 5b shows several RL calibrations obtained by couopling the RL framework described in the main text with the following MAB learning algorithms, all available through the *SMPyBandits* package (Besson 2018): 'kl-UCB' (Garivier and Cappé 2011), 'Exp3' (Bubeck, Cesa-Bianchi et al. 2012), 'Exp3.R' (Allesiardo and Féraud 2015) and 'Exp3++' (Seldin and Slivkins 2014), 'Thompson' and 'Discounted Thompson' sampling (Raj and Kalyani 2017). All of the learning schemes achieve satisfactory results by outperforming all sub-optimal samplers, and performing on par with the RF sampler, but without any prior information about the best sampler at the agent's disposal. Figure 5c shows the RL calibration obtained using the ϵ-greedy scheme proposed and tested also in the main text, for different choices of ϵ and α. The ϵ-greedy scheme is also seen to outperform all single samplers of Figure 5a except the optimal one, and its performance is found to be very similar to those of the other algorithms tested in Figure 5c.

Figure 5d depicts the actions selected during the 5 runs pertaining to the ϵ-greedy calibration with $\epsilon = \alpha = 0.1$. Some patterns are clearly visible, such as the preferential choice of the BB sampler and the RF sampler, particularly in the first half of the calibration where the loss decreases rapidly before reaching a plateau.

B Data and code availability

In the interest of reproducibility, the code, the data and the scripts used to generate the key results and the main graphs of this work are available to download as supplementary material of the paper.