JOSS
The Journal of Open Source Software

Black-it: A Ready-to-Use and Easy-to-Extend Calibration Kit for Agent-based Models

Marco Benedetti[1], **Gennaro Catapano**[1], **Francesco De Sclavis**[1], **Marco Favorito**[1], **Aldo Glielmo**[1], **Davide Magnanimi**[1], **and Antonio Muci**[1]

1 Banca d'Italia, Italy

DOI: 10.21105/joss.04622

Software

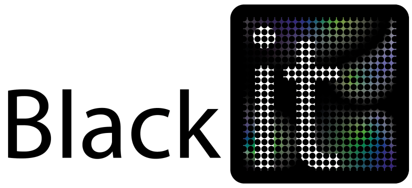
- Review
- Repository
- Archive

Editor: Vincent Knight

Reviewers:

- @Athene-ai
- @aseyq

Submitted: 01 July 2022
Published: 03 November 2022



Black-**it** logo.

Summary

We present *Black*-**it**, a software package that aims to collect and popularise recent developments in the field of agent-based model calibration, and to provide an extensible testbed for future experimentation. We showcase its capabilities by benchmarking different calibration methods on standard calibration tasks, and find that mixing calibration strategies often improves performance.

Statement of need

Agent-based models (ABMs) are an increasingly common technique for describing and understanding emergent phenomena in complex social and economic systems (Delli Gatti et al., 2018; Miller et al., 2008; Richiardi, 2012; Turrell, 2016).

Over the past decades, ABMs evolved from very simple models involving a few hundred agents moving on two-dimensional grids (Schelling, 1971) to large-scale realistic simulations with tens of thousands of heterogeneous agents. Notable applications range from modelling the housing market (Baptista et al., 2016; Catapano et al., 2021) to understanding the spreading of COVID-19 (Hinch et al., 2021).

In order to profit from ABMs, it is necessary to *calibrate* them, i.e., to find values for the set of parameters in the model that make it best fit real-world evidence and historical data available for the target phenomenon.

Unfortunately, the increasing scale and complexity of current models makes their calibration a significant challenge (Platt, 2020), as any non-trivial model has dozens of continuous and discrete parameters, which generate an enormous search space. Its effective exploration by brute-force or trial-and-error approaches (Tisue & Wilensky, 2004) is impractical.

The vastness of the parameter space of state-of-the-art ABMs calls for sophisticated search and calibration strategies. Research in this direction has evolved significantly in recent years, and many competing calibration techniques have been proposed (Dyer et al., 2022; Grazzini et al., 2017; Lamperti et al., 2018; Platt, 2021; Ward et al., 2016). However, they have been tested only in specific scenarios and remain out of reach for most researchers interested in ABM *applications*.

In this work we address such shortcoming by introducing *Black*-**it**[1], a calibration kit specifically designed for ABMs. Our goal is to help ABM researchers from different disciplines to conveniently calibrate their models by gathering, within a powerful and easy-to-use tool, several recent advances in the field. The software is designed to be easy to extend, so it doubles as a research testbed for experimenting with new or improved ideas.

Black-**it** differs from standard optimisation packages available in the Python ecosystem, such as SciPy (Virtanen et al., 2020), in three main aspects. First, *Black*-**it** not only allows the use of a set of advanced optimisation algorithms, but also provides a general framework where new calibration schemes can easily be implemented and released to the ABM community. Second, optimisation algorithms in *Black*-**it** are specifically designed for ABMs, i.e., for non-convex, highly nonlinear and often discontinuous functions with no computable gradients, and cannot be found elsewhere collected in a single software. Third, *Black*-**it** includes a rich set of features that make its use particularly convenient for ABM researchers, such as a set of standard loss functions, parallelisation and checkpointing, plotting functionalities, and more.

The defining features of our tool will be reviewed in the next sections, where we also showcase its ease of use and capabilities by benchmarking a set of complex calibration methods on some standard calibration tasks, using just a few lines of code[2].

Software description

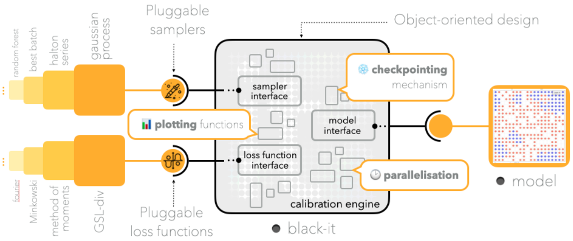


Figure 1: Illustration of the main features of *Black*-**it**.

```
from black_it import Calibrator
from black_it.samplers import (HaltonSampler,
                               RandomForestSampler)
from black_it.loss_functions import
 MethodOfMomentsLoss
# initialise samplers to take 10 samples each
halton = HaltonSampler(10)
random_forest = RandomForestSampler(10)
# initialise loss function
loss = MethodOfMomentsLoss()
# initialise calibrator
cal = Calibrator(
      samplers=[halton, random_forest],
      real_data=real_data,
      model=model,
      parameters_bounds=bounds,
      parameters_precision=precisions,
      loss_function=loss)
# calibrate for 30 epochs
params, losses = cal.calibrate(30)
```

Figure 2: A simple *Black*-**it** calibration script.

At a high level, *Black*-**it** calibration works as follows.

First, a *sampling method* (or *sampler*) is summoned to suggest a set of promising parameter configurations to explore. Second, the model to be calibrated is simulated for all the selected parameters. Third, a specific *loss function*, measuring the “fitness” of the simulations with respect to the real data, is evaluated. These three steps are performed in a loop, and this allows the samplers to progress towards better parameter values by exploiting the knowledge of previously computed loss functions. The calibration loop ends when a certain target loss is achieved, or when a maximum number of epochs is reached.

Black-**it** implements a number of state-of-the-art samplers, including low-discrepancy sampling series (Knysh & Korkolis, 2016) as well as adaptive samplers exploiting machine learning surrogates (Lamperti et al., 2018) or genetic algorithms (Stonedahl, 2011). Moreover, our tool

[1] *Black*-**it** is an acronym for **Bl**ack-box **A**BM calibration **kit** (by the Bank of **It**aly).
[2] Visit https://github.com/bancaditalia/black-it/blob/main/examples/benchmarking_samplers.ipynb for an example script.

enables the construction of hybrid samplers by combining together multiple base samplers, a strategy that often improves performance, as we show in the next section. In addition, our tool implements a number of standard loss functions, such as the "method of moments" distance (Franke, 2009) and the GSL-divergence (Lamperti, 2018). Currently, only likelihood-free losses are implemented, since probabilistic methods are typically too expensive to be used in large-scale ABMs (Platt, 2020), but nothing hinders the inclusion of the latest advancements in the field (Dyer et al., 2022; Platt, 2021) into our tool.

Black-**it** provides a range of "convenience" features, such as (i) efficient on-demand parallelisation, (ii) a checkpointing mechanism to robustly persist the calibration state, and (iii) plotting functions to visualise ongoing and completed calibrations.

The toolkit is written in Python and it has a modular and object-oriented design, features that make it seamless to implement custom sampling methods and loss functions, if needed. Figure 1 provides a visual summary of the main features of the package, while Figure 2 shows a basic calibration script.

Benchmarking of sampling methods

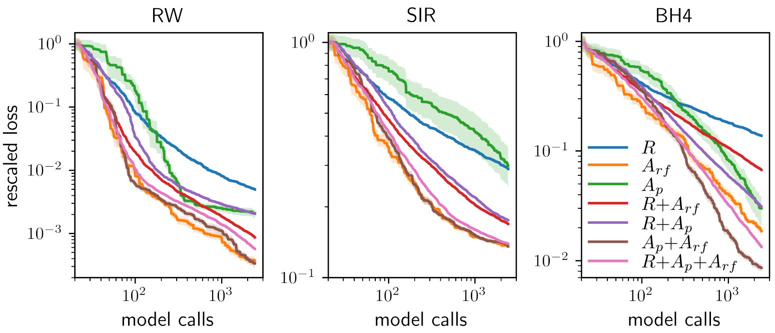


Figure 3: Loss as a function of the number of model evaluations for 3 models and 7 types of sampling strategies.

We showcase the capabilities of *Black*-**it** by benchmarking the performance of a set of sampling methods on the calibration of three standard models.

Models

We consider the following models: (i) A random walk model ('RW') with a structural break (Platt, 2020) (5 free parameters), (ii) a SIR model on a Watts-Strogatz network (Simoes et al., 2008; Watts & Strogatz, 1998) (5 free parameters) (iii) the standard Brock and Hommes model for asset pricing with 4 belief types ('BH4') (Brock & Hommes, 1998) (8 free parameters).

For (i) and (iii) we use a method of moment distance loss, while for (ii) we use an Euclidean distance loss. The target ('real') time series for the calibration were obtained by simulating the models with reasonable parameter choices.

Sampling methods

We consider two types of samplers: random and adaptive. Adaptive samplers propose parameters on the basis of previously computed loss values, while random samplers do not use that information and propose parameters in a purely random fashion.

RW	SIR	BH4
$A_p + A_{rf}$	$A_p + A_{rf}$	$A_p + A_{rf}$
A_{rf}	A_{rf}	$R_h + A_p + A_{rf}$
$R_u + A_p + A_{rf}$	$R_h + A_p + A_{rf}$	$R_u + A_p + A_{rf}$

Table 1: Top 3 sampling strategies for each model considered.

As adaptive samplers (A) we consider a random-forest surrogate of the loss function (A_{rf}) (Bajer et al., 2015; Lamperti et al., 2018), and a genetic-type algorithm (Stonedahl, 2011) that samples small perturbations around the current optimal parameters (A_p).

Experimental procedure

For each model, and fixing the number of model calls to 2400 (100 epochs with a batch size of 24), we perform calibration runs for the 5 samplers described, as well as for all 80 possible combinations of 2 or 3 samplers together. We average all results over 15 repetitions.

Results

In Figure 3 we report the minimal loss as a function of the number of calls for the 3 models, aggregated for different groups of sampling strategies. Not surprisingly, the non-adaptive (purely random) samplers R generally demonstrate a very low performance (blue curve). Similarly, the small perturbation sampler A_p is also seen to underperform (green line) but, interestingly, the performance of both R and A_p improves significantly as the two approaches are combined into a mixed strategy $R + A_p$ (purple curve), striking a more balanced *exploration-exploitation* trade-off. Differently from A_p, the random forest sampler A_{rf} performs very well on its own (orange curve), while its performance is deteriorated by the addition of random samplers (red curve). However, the performance of A_{rf} is surpassed by the mixed strategy $A_p + A_{rf}$ (brown curve), especially in late calibration epochs. The 3-sampler combination $R + A_p + A_{rf}$ provides generally competitive results, as confirmed by Table 1, which reports the best strategies for each model.

Acknowledgements

The authors thank Andrea Gentili (Banca d'Italia) for the design of Figure 1 and of all *Black*-**it** schematics, and Sara Corbo (Banca d'Italia) for the design of the *Black*-**it** logo.

References

Bajer, L., Pitra, Z., & Holeňa, M. (2015). Benchmarking Gaussian processes and random forests surrogate models on the BBOB noiseless testbed. *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 1143–1150. https://doi.org/10.1145/2739482.2768468

Baptista, R., Farmer, J. D., Hinterschweiger, M., Low, K., Tang, D., & Uluc, A. (2016). *Macroprudential policy in an agent-based model of the UK housing market*. https://doi.org/10.2139/ssrn.2850414

Brock, W. A., & Hommes, C. H. (1998). Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, *22*(8-9), 1235–1274. https://doi.org/10.1016/S0165-1889(98)00011-6

Catapano, G., Franceschi, F., Loberto, M., & 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*. https://doi.org/10.2139/ssrn.3891583

Delli Gatti, D., Fagiolo, G., Gallegati, M., Richiardi, M., & Russo, A. (2018). *Agent-based models in economics: A toolkit*. Cambridge University Press.

Dyer, J., Cannon, P., Farmer, J. D., & Schmon, S. (2022). Black-box Bayesian inference for economic agent-based models. *arXiv Preprint arXiv:2202.00625*. https://doi.org/10.48550/arXiv.2202.00625

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. https://doi.org/10.1016/j.jempfin.2009.06.006

Grazzini, J., Richiardi, M. G., & Tsionas, M. (2017). Bayesian estimation of agent-based models. *Journal of Economic Dynamics and Control*, *77*, 26–47. https://doi.org/10.1016/j.jedc.2017.01.014

Hinch, R., Probert, W. J., Nurtay, A., Kendall, M., Wymant, C., Hall, M., Lythgoe, K., Bulas Cruz, A., Zhao, L., Stewart, A., & others. (2021). OpenABM-Covid19—an agent-based model for non-pharmaceutical interventions against COVID-19 including contact tracing. *PLoS Computational Biology*, *17*(7), e1009146. https://doi.org/10.1371/journal.pcbi.1009146

Knysh, P., & Korkolis, Y. (2016). Blackbox: A procedure for parallel optimization of expensive black-box functions. *arXiv Preprint arXiv:1605.00998*. https://doi.org/10.48550/arXiv.1605.00998

Lamperti, F. (2018). An information theoretic criterion for empirical validation of simulation models. *Econometrics and Statistics*, *5*, 83–106. https://doi.org/10.1016/j.ecosta.2017.01.006

Lamperti, F., Roventini, A., & Sani, A. (2018). Agent-based model calibration using machine learning surrogates. *Journal of Economic Dynamics and Control*, *90*, 366–389. https://doi.org/10.1016/j.jedc.2018.03.011

Miller, J. H., Page, S. E., & LeBaron, B. (2008). Complex adaptive systems: An introduction to computational models of social life. *Journal of Economic Literature*, *46*(2), 427–428.

Platt, D. (2020). A comparison of economic agent-based model calibration methods. *Journal of Economic Dynamics and Control*, *113*, 103859. https://doi.org/10.1016/j.jedc.2020.103859

Platt, D. (2021). Bayesian estimation of economic simulation models using neural networks. *Computational Economics*, 1–52. https://doi.org/10.1007/s10614-021-10095-9

Richiardi, M. G. (2012). Agent-based computational economics: A short introduction. *The Knowledge Engineering Review*, *27*(2), 137–149.

Schelling, T. C. (1971). Dynamic models of segregation. *Journal of Mathematical Sociology*, *1*(2), 143–186. https://doi.org/10.1080/0022250X.1971.9989794

Simoes, M., Telo da Gama, M., & Nunes, A. (2008). Stochastic fluctuations in epidemics on networks. *Journal of the Royal Society Interface*, *5*(22), 555–566. https://doi.org/10.1098/rsif.2007.1206

Stonedahl, F. J. (2011). *Genetic algorithms for the exploration of parameter spaces in agent-based models* [PhD thesis]. Northwestern University.

Tisue, S., & Wilensky, U. (2004). Netlogo: A simple environment for modeling complexity. *International Conference on Complex Systems*, *21*, 16–21.

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

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Ward, J. A., Evans, A. J., & Malleson, N. S. (2016). Dynamic calibration of agent-based models using data assimilation. *Royal Society Open Science*, *3*(4), 150703. https://doi.org/10.1098/rsos.150703

Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, *393*(6684), 440–442. https://doi.org/10.1038/30918