Some challenges of calibrating differentiable agent-based models

Arnau Quera-Bofarull [*1] **Joel Dyer** [*1 2] **Anisoara Calinescu** [1] **Michael Wooldridge** [1]

Abstract

Agent-based models (ABMs) are a promising approach to modelling and reasoning about complex systems, yet their application in practice is impeded by their complexity, discrete nature, and the difficulty of performing parameter inference and optimisation tasks. This in turn has sparked interest in the construction of differentiable ABMs as a strategy for combatting these difficulties, yet a number of challenges remain. In this paper, we discuss and present experiments that highlight some of these challenges, along with potential solutions.

1. Introduction

Agent-based models (ABMs, see Appendix A for a brief overview) have gained considerable popularity across a range of disciplines, due to their ability to accurately simulate complex systems at a granular level. While these models offer unique advantages, their complexity presents significant challenges, for example in terms of parameter calibration (see e.g. Dyer et al., 2022a;b). For such tasks, multiple factors contribute to their difficulty, including the intractability of the ABM's likelihood function, and the often black-box and non-differentiable nature of the ABM.

These drawbacks of ABMs have motivated research into the construction of differentiable ABMs (Chopra et al., 2023; Quera-Bofarull, 2023), for example through the use of differentiable programming and by exploiting automatic differentiation (AD) frameworks. AD – a methodological cornerstone in machine learning, largely underpinning the success of deep learning paradigms due to its ability to accurately compute derivatives within models – circumvents issues present in alternative approaches to model differentiation by applying the chain rule of differentiation at a computational level, resulting in exact derivatives.

Despite recent progress, the challenges involved in building and benefitting from differentiable ABMs remain underexplored, and there exists little guidance to practitioners interested in implementing and exploiting differentiable ABMs. The aim of this paper is therefore to discuss some central challenges in applying AD to ABMs.

2. Challenges

2.1. Discrete randomness

The issue of differentiating through discrete structures is inherent in ABMs, which simulate discrete events, transitions, and interactions that are incompatible with conventional AD. Initial efforts to implement AD within ABMs have primarily centred on transforming the ABMs' discrete control flow structure with continuous approximations (Andelfinger, 2021). Furthermore, the use of the Gumbel-Softmax (GS) reparametrisation trick (Jang et al., 2017) allows for the differentiation of discrete randomness, and has been deployed effectively in epidemiological ABMs (Chopra et al., 2023). However, this approach does not provide an ideal solution. While it allows for gradient calculations, GS does not guarantee unbiased or low-variance gradients (Huijben et al., 2022). Developing unbiased and lower variance methods such as StochasticAD (Arya et al., 2022) in the Julia programming language (Bezanson et al., 2017) is currently an active field of research, but we limit the scope of our discussion here to GS-based methods for discrete ABMs.

Despite the potential lack of robustness of GS, GS-based differentiable ABM implementations have shown great success in improving the calibration (Chopra et al., 2023) and sensitivity analyses (Quera-Bofarull et al., 2023) of ABMs. In the Experiments section below, we further show that gradients obtained using the GS trick are robust enough to enable fast and accurate Bayesian inference (see subsection 3.2).

2.2. Reverse- vs. Forward-mode AD

In Reverse-mode AD (RMAD), a computation graph must be stored that records all operations performed within the model, such that the gradients of the model outputs with respect to the input parameters can be obtained. This contrasts with Forward-mode AD (FMAD), where the gradients

[*]Equal contribution [1]Department of Computer Science, University of Oxford [2]Institute for New Economic Thinking, Oxford. Correspondence to: Arnau Quera-Bofarull <arnau.quera-bofarull@cs.ox.ac.uk>, Joel Dyer <joel.dyer@cs.ox.ac.uk>.

Published at the Differentiable Almost Everything Workshop of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. July 2023. Copyright 2023 by the author(s).

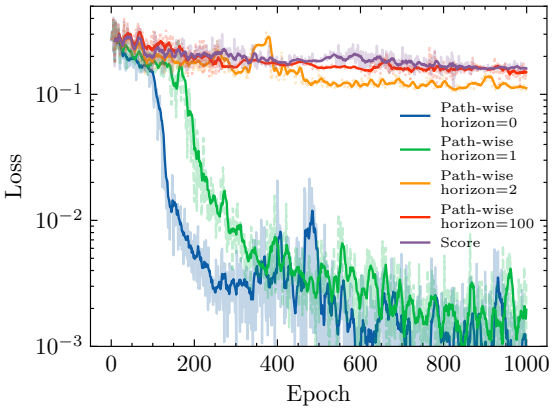

Figure 1. Training loss as a function of epochs for each gradient estimation method and gradient horizon value. Solid line shows 10-step moving average; dashed lines are without averaging.

are computed during the forward simulation. There are two important computational considerations when comparing FMAD vs. RMAD. The first is that the computational time associated with FMAD scales with the number of model inputs, while that of RMAD scales with the number of model outputs. In machine learning, the latter option is more prevalent, since machine learning models often have many more inputs than outputs. However, the computation graph that RMAD must store in (often GPU) memory can be extremely large, hindering the possibility of differentiating through large models. This is particularly pertinent for ABMs: the size of the computation graph grows with the number of agents and time-steps, which can pose a challenge to the use of RMAD for ABMs with a large number of agents and time-steps.

To address this, in subsection 3.2 we discuss a differentiating strategy that alternates between FMAD and RMAD when calibrating ABMs, and we apply it to an epidemiological simulation involving 8 million agents.

2.3. Monte Carlo gradient estimation

Since ABMs are typically stochastic models, it can often be the case that practitioners are interested in performing an optimisation problem of the form

$$\min_{\omega \in \Omega} \mathbb{E}_{z \sim p_\omega} \left[\mathcal{L}(z) \right], \tag{1}$$

where $p_\omega \in \{p_{\omega'} : \omega' \in \Omega\}$ is a probability distribution on some domain $\mathcal{Z}$ indexed by a parameter ω belonging to some set Ω, and $\mathcal{L} : \mathcal{Z} \to \mathbb{R}$ is a loss function. For example, certain parameter calibration procedures that seek to identify the parameters $\boldsymbol{\theta}$ in some set $\boldsymbol{\Theta}$ that minimise some discrepancy $\mathcal{D}(\cdot, \mathbf{y})$ between the model output $\mathbf{x}$ and real-world data $\mathbf{y}$ can be cast in the form

$$\min_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathbb{E}_{\mathbf{x} \sim p(\cdot \mid \boldsymbol{\theta})} \left[\mathcal{D}(\mathbf{x}, \mathbf{y}) \right], \tag{2}$$

where $p(\cdot \mid \boldsymbol{\theta})$ is the ABM's likelihood function. The gradients of a differentiable ABM can then be exploited by gradient-assisted methods for minimising the objective in (1), by finding a Monte Carlo estimate of the expression

$$\nabla_\omega \mathbb{E}_{z \sim p_\omega} \left[\mathcal{L}(z) \right]. \tag{3}$$

For differentiable ABMs, a Monte Carlo estimate of this gradient can be obtained using the path-wise derivative via reparametrisation tricks (Mohamed et al., 2020). In such cases, derivatives of the form $\partial \mathbf{x}_t / \partial \omega_i$ will contribute to the estimate. To properly benefit from access to the differentiable ABM's gradients in these settings, it is critical that low-variance, low-bias Monte Carlo estimates of (3) are available. However, as we will demonstrate in subsection 3.1, naively estimating these gradients by accounting for both (a) the explicit dependency of each $\mathbf{x}_t$ on ω_i, and (b) the implicit dependency on ω_i, mediated by the $\mathbf{x}_{1:t-1}$ that result from the recursive structure of ABMs, can result in unusable gradient estimates with prohibitively large variances. Consequently, modifications to vanilla AD can become necessary, as illustrated in subsection 3.1.

3. Experiments

In this section, we present experiments on the use of gradient-assisted calibration methods for two ABMs, where each experiment serves to highlight different combinations of the challenges described in Section 2.

While there exist many different gradient-assisted calibration methods, we focus on a variational approach to Bayesian parameter inference termed Generalised Variational Inference (GVI, Knoblauch et al., 2022) – a likelihood-free Bayesian inference approach that has previously been used to calibrate the parameters $\boldsymbol{\theta} \in \mathbb{R}^d$ of a differentiable ABM (Quera-Bofarull et al., 2023). Here, a variational procedure targets a "generalised" posterior (Bissiri et al., 2016)

$$\pi_{w,\mathbf{y}}(\boldsymbol{\theta}) \propto e^{-w \cdot \ell(\mathbf{y}, \boldsymbol{\theta})} \pi(\boldsymbol{\theta}), \tag{4}$$

where $\pi(\boldsymbol{\theta})$ is a prior distribution, $\ell(\mathbf{y}, \boldsymbol{\theta})$ is a loss function capturing the compatibility between the observed data $\mathbf{y}$ and the behaviour of the ABM at parameter vector $\boldsymbol{\theta}$, and $w > 0$ is a hyperparameter. To target this posterior, we train a normalising flow q_ϕ with trainable parameters ϕ to minimise the Kullback-Liebler divergence $\mathrm{KL}(q_\phi \| \pi_{w,\mathbf{y}})$ from q_ϕ to $\pi_{w,\mathbf{y}}$, yielding the minimising parameters

$$\phi_{w,\mathbf{y},\pi} = \arg\min_{\phi} \left\{ w \, \mathbb{E}_{q_\phi} \left[\ell(\mathbf{y}, \boldsymbol{\theta}) \right] + \mathrm{KL}\left(q_\phi \| \pi \right) \right\}. \tag{5}$$

Further details are provided in Appendix D. Code to reproduce the results and perform GVI on differentiable ABMs can be found at `https://github.com/arnauqb/blackbirds`.

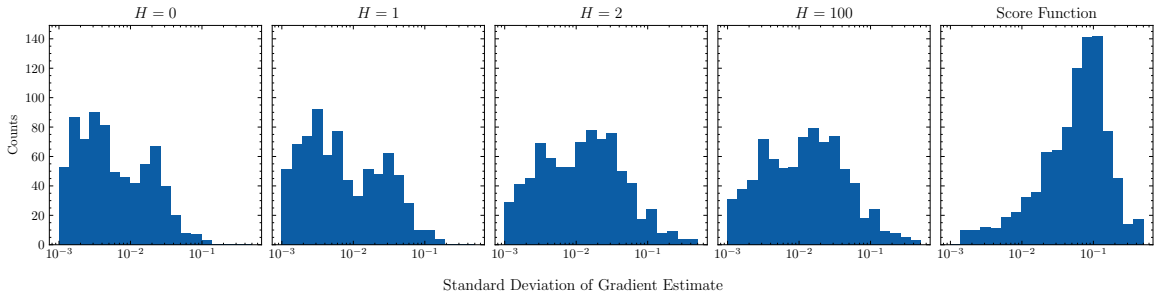

Figure 2. Histograms of the standard deviations in estimates of $\partial \mathbb{E}_{q_\phi} \ell(\mathbf{y}, \boldsymbol{\theta}) / \partial \phi_j$ for different gradient horizons and the score-based method across all j for the Brock & Hommes model.

3.1. The Brock & Hommes model

The Brock & Hommes model (Brock & Hommes, 1998) is a heterogeneous agent model for the price $\mathbf{x}_t \in \mathbb{R}$ of an asset over time $1 \leq t \leq T$. At each time step, the agents in the model subscribe to one of a set of $J > 1$ trading strategies, each of which is characterised by a trend-following parameter g_j and bias parameter b_j, $j \in \{1, \ldots, J\}$. Following Dyer et al. (2022a), we note that the price $\mathbf{x}_t$ may be written as deterministic transformations f_t of the input parameters $\boldsymbol{\theta} = (g_1, \ldots, g_J, b_1, \ldots, b_J)$, auxiliary parameters $\boldsymbol{\alpha}$, and standard Normal random variables:

$$\mathbf{x}_t = f_t(\epsilon_1, \ldots, \epsilon_t, \boldsymbol{\theta}, \boldsymbol{\alpha}), \quad \epsilon_t \sim \mathcal{N}(0, 1). \tag{6}$$

Further details are provided in Appendix B. Thus, provided ℓ is chosen to be a differentiable function of $\boldsymbol{\theta}$, this enables us to employ gradient-based approaches to minimising the objective (5) that exploit the reparameterisation trick.

Fixing $g_1 = b_1 = b_4 = 0$ and $g_4 = 1.01$, we consider the task of calibrating parameters g_2, g_3, b_2, b_3 given synthetic data $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ generated from the model at $(g_2, g_3, b_2, b_3) = (0.9, 0.9, 0.2, -0.2)$ with $T = 100$. We follow Cherief-Abdellatif & Alquier (2020) and target the generalised posterior (4) given by the choice

$$\ell(\mathbf{y}, \boldsymbol{\theta}) = \mathrm{MMD}^2(\mathbb{P}_T, \mathbb{P}_{\boldsymbol{\theta}}), \tag{7}$$

where $\mathrm{MMD}^2(\mathbb{P}_T, \mathbb{P}_{\boldsymbol{\theta}})$ is the maximum mean discrepancy between the empirical measure of returns $\mathbb{P}_T = (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ and the distribution $\mathbb{P}_{\boldsymbol{\theta}}$ of returns implied by the simulator at parameters $\boldsymbol{\theta}$. Using a Gaussian kernel within the MMD computation, the operations comprising evaluation of $\ell(\mathbf{y}, \boldsymbol{\theta})$ are also all differentiable and deterministic, enabling evaluation of the term $\nabla_\phi \mathbb{E}_{\boldsymbol{\theta} \sim q_\phi} \ell(\mathbf{y}, \boldsymbol{\theta})$ in (5) using the reparameterisation trick (see Appendix B.1).

Despite our ability to compute the partial derivatives $\partial \mathbf{x}_t / \partial \phi_i$ exactly, the posterior estimator q_ϕ struggles to train with a gradient-assisted approach to minimising (5). This can be seen in Figure 1, in which the objective function decreases slowly with the number of epochs when trained with AdamW (Loshchilov & Hutter, 2017) and using the vanilla pathwise derivative (red curve). Indeed, we see in this case that the access to the simulator's gradients appears to offer no improvement over the score-based gradient, shown with the purple curve and obtained as

$$\nabla_\phi \mathbb{E}_{q_\phi} \left[\ell(\mathbf{y}, \boldsymbol{\theta}) \right] = \mathbb{E}_{q_\phi} \left[\ell(\mathbf{y}, \boldsymbol{\theta}) \nabla_\phi \log q_\phi(\boldsymbol{\theta}) \right]. \tag{8}$$

Drawing inspiration from the literature on backpropagation-through-time (e.g. truncated back-propagation in the context of RNN training, see Sutskever (2013)), we consider pruning a subset of the paths in the computation graph that contribute to each of the $\partial \mathbf{x}_t / \partial \boldsymbol{\theta}_i$ as a possible solution to this problem. We achieve this by invoking an appropriate `stop_gradient` operation (e.g. `.detach()` in `pytorch`, Paszke et al. (2019)) on terms $\mathbf{x}_{t'}$ that (a) contribute directly/explicitly to the evaluation of $\mathbf{x}_t$ and (b) for which $t' < t - H$ for some "gradient horizon" $H \geq 0$. As evident from Figure 1, we observe that a finite gradient horizon can dramatically improve the gradient-assisted training of q_ϕ. In this experiment, the best performance was observed while using a gradient horizon of $H = 0$.

We posit that this is a manifestation of a bias-variance trade-off in the Monte Carlo gradient estimation step: pruning a subset of paths in the computation graph with the use of a finite gradient horizon may introduce some bias in, but can significantly reduce the variance of, Monte Carlo estimates of the gradient $\nabla_\phi \mathbb{E}_{q_\phi} \left[\ell(\mathbf{y}, \boldsymbol{\theta}) \right]$ when employing the pathwise derivative. This hypothesis is supported by Figure 2, which shows histograms of the standard deviation of the estimates of $\partial \mathbb{E}_{q_\phi} \left[\ell(\mathbf{y}, \boldsymbol{\theta}) \right] / \partial \phi_j$ across all j for gradient horizons $H \in \{0, 1, 2, 100\}$, and for the score-based estimator. There, we see that the histogram shifts towards larger values as H increases. Further results supporting this hypothesis are given in Appendix B.2.

3.2. The JUNE model

The JUNE model (Aylett-Bullock et al., 2020) is a large-scale epidemiological ABM of England based on a realistic synthetic population constructed from the English census. Calibrating the original implementation required the construction of a surrogate model due to its high computational

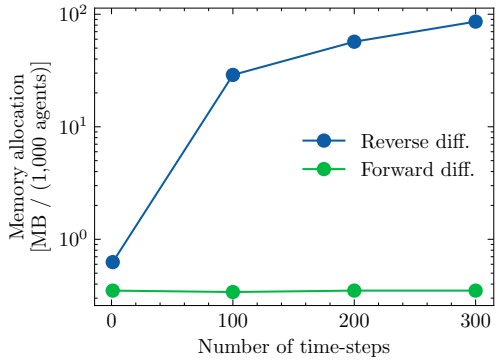

Figure 3. Memory consumption of running AD in the JUNE model per 1,000 agents, for the Forward and Reverse mode of AD.

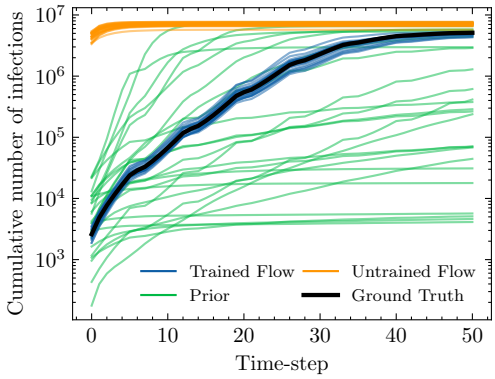

Figure 4. Cumulative infections over time. Black: synthetic ground truth; green, orange, blue: runs obtained from samples from the prior, untrained flow, and trained flow, respectively.

cost (Vernon et al., 2022). The GRADABM-JUNE model (Quera-Bofarull, 2023) is a differentiable implementation of JUNE which employs the GS reparameterisation trick to differentiate through discrete randomness. Compared to its non-differentiable counterpart, GRADABM-JUNE has been used to more efficiently generate parameter point estimates, as well as sensitivity analyses (Quera-Bofarull et al., 2023).

3.2.1. REDUCING MEMORY CONSUMPTION THROUGH FORWARD-MODE AD

GRADABM-JUNE can simulate the entire English population at a scale of 1:1 — 53 million people at the time of the 2011 census. As discussed in subsection 2.2, differentiating through this model using RMAD is challenging due to the high memory demand to store the computation graph. To perform GVI for this model, we implement a hybrid AD technique: we use FMAD to obtain the Jacobian $J_{\boldsymbol{\theta}}$ of the ABM outputs with respect to the ABM parameters, and combine it with RMAD through the flow q_ϕ, yielding

$$\nabla_\phi \mathbb{E}_{q_\phi}\left[\ell(\mathbf{y}, \boldsymbol{\theta})\right] = J_{\boldsymbol{\theta}}(\mathbb{E}_{q_\phi}\left[\ell(\mathbf{y}, \boldsymbol{\theta})\right]) \cdot \nabla_\phi \boldsymbol{\theta}, \quad \text{with} \tag{9}$$

$$J_{\boldsymbol{\theta}}(\mathbb{E}_{q_\phi}\ell(\mathbf{y}, \boldsymbol{\theta})) = \frac{\partial \mathbb{E}_{q_\phi}\ell(\mathbf{y}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{1\times d}. \tag{10}$$

Here, (10) is the Jacobian obtained through FMAD and $\nabla_\phi \boldsymbol{\theta} \in \mathbb{R}^{d\times F}$ is the gradient, obtained with RMAD, of the d ABM parameters generated by the normalising flow with parameters $\phi \in \mathbb{R}^F$.

In Figure 3, we plot the memory costs of employing FMAD and RMAD to compute the ABM's Jacobian. We see that the cost of FMAD is independent of the number of time-steps, since no computation graph is stored. In contrast, the cost of RMAD scales linearly with the number of time-steps and agents. Simulating the entire English population for 300 time-steps would require 5TB of memory, while doing so with FMAD would require merely 18GB, regardless of the number of time-steps. The increase in computational time of FMAD comes at an increase of computational cost since it requires d evaluations of the model for $\boldsymbol{\theta} \in \mathbb{R}^d$; however, since these evaluations are embarrassingly parallelisable, the impact on performance can be minimal.

With the above in mind, we set up an experiment with the London's population (8.1 million people) in GRADABM-JUNE. We generate a synthetic time-series of daily infections for 50 days using some assumed parameters that we aim to recover through our calibration process. The parameters that we vary are the contact intensities at 10 different locations, as well as the number of initial cases. Further details of the experimental setup are shown in Appendix C.

We apply the GVI procedure to calibrate the JUNE model with 11 free parameters. The flow converges after approximately 3,000 model evaluations, highlighting the potential for simulation-efficient calibration with gradient-assisted methods. In Figure 4, we show a comparison of runs obtained by sampling the ABM parameters from the trained flow, the untrained flow, and the prior. This demonstrates that the trained flow generates parameters that result in close agreement between the simulator and ground truth data, while providing useful uncertainty quantification.

4. Conclusions and discussion

This study examines some challenges that arise from the application of vanilla AD to ABMs, such as overcoming the inherent discreteness of ABMs and the variance and high computational requirements of passing gradients through large simulators. We have shown that these challenges can be overcome to some extent with different modifications to vanilla AD. As supporting evidence, we successfully calibrate differentiable implementations of the Brock & Hommes and JUNE models with these modifications, the latter involving over eight million agents and discrete randomness. In this way, this study helps to pave the way towards robust calibration of large-scale agent-based models.

References

Andelfinger, P. Differentiable Agent-Based Simulation for Gradient-Guided Simulation-Based Optimization. *arXiv:2103.12476 [cs, eess]*, March 2021.

Arya, G., Schauer, M., Schäfer, F., and Rackauckas, C. Automatic Differentiation of Programs with Discrete Randomness, October 2022.

Aylett-Bullock, J., Cuesta-Lazaro, C., Quera-Bofarull, A., Icaza-Lizaola, M., Sedgewick, A., Truong, H., Curran, A., Elliott, E., Caulfield, T., Fong, K., Vernon, I., Williams, J., Bower, R., and Krauss, F. June: Open-source individual-based epidemiology simulation. *Royal Society Open Science*, 8(7):210506, 2020. doi: 10.1098/rsos.210506.

Aylett-Bullock, J., Cuesta-Lazaro, C., Quera-Bofarull, A., Katta, A., Pham, K. H., Hoover, B., Strobelt, H., Jimenez, R. M., Sedgewick, A., Evers, E. S., Kennedy, D., Harlass, S., Maina, A. G. K., Hussien, A., and Luengo-Oroz, M. Operational response simulation tool for epidemics within refugee and IDP settlements: A scenario-based case study of the Cox's Bazar settlement. *PLOS Computational Biology*, 17(10):e1009360, October 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1009360.

Baptista, R., Farmer, J. D., Hinterschweiger, M., Low, K., Tang, D., and Uluc, A. Staff working paper no. 619 macroprudential policy in an agent-based model of the uk housing market. 2016.

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98, 2017.

Bissiri, P. G., Holmes, C., and Walker, S. G. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103, 2016.

Brock, W. A. and Hommes, C. H. Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *Journal of Economic Dynamics and Control*, 22(8):1235–1274, July 1998. ISSN 0165-1889. doi: 10.1016/S0165-1889(98)00011-6.

Cherief-Abdellatif, B.-E. and Alquier, P. MMD-Bayes: Robust Bayesian Estimation via Maximum Mean Discrepancy. In Zhang, C., Ruiz, F., Bui, T., Dieng, A. B., and Liang, D. (eds.), *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, volume 118 of *Proceedings of Machine Learning Research*, pp. 1–21. PMLR, 08 Dec 2020. URL https://proceedings.mlr.press/v118/cherief-abdellatif20a.html.

Chopra, A., Rodríguez, A., Subramanian, J., Quera-Bofarull, A., Krishnamurthy, B., Prakash, B. A., and Raskar, R. Differentiable agent-based epidemiology. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pp. 1848–1857, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-9432-1.

Dyer, J., Cannon, P., Farmer, J. D., and Schmon, S. Black-box Bayesian inference for economic agent-based models. *arXiv preprint arXiv:2202.00625*, 2022a.

Dyer, J., Cannon, P., Farmer, J. D., and Schmon, S. M. Calibrating Agent-based Models to Microdata with Graph Neural Networks. In *ICML 2022 Workshop AI for Agent-Based Modelling*, 2022b.

Huijben, I. A. M., Kool, W., Paulus, M. B., and van Sloun, R. J. G. A Review of the Gumbel-max Trick and its Extensions for Discrete Stochasticity in Machine Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1353–1371, 2022.

Jang, E., Gu, S., and Poole, B. Categorical Reparameterization with Gumbel-Softmax. *arXiv:1611.01144 [cs, stat]*, August 2017.

Knoblauch, J., Jewson, J., and Damoulas, T. An optimization-centric view on bayes' rule: Reviewing and generalizing variational inference. *Journal of Machine Learning Research*, 23(132):1–109, 2022.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. Monte Carlo Gradient Estimation in Machine Learning. *The Journal of Machine Learning Research*, 21(1):5183–5244, 2020.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Quera-Bofarull, A. Arnauqb/GradABM-JUNE: AAMAS zenodo release. Zenodo, February 2023.

Quera-Bofarull, A., Chopra, A., Aylett-Bullock, J., Cuesta-Lazaro, C., Calinescu, A., Raskar, R., and Wooldridge, M. Don't simulate twice: One-shot sensitivity analyses via automatic differentiation. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, pp. 1867–1876, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-9432-1.

Quera-Bofarull, A., Chopra, A., Calinescu, A., Wooldridge, M., and Dyer, J. Bayesian calibration of differentiable agent-based models. *ICLR Workshop on AI for Agent-based Modelling*, 2023.

Stimper, V., Liu, D., Campbell, A., Berenz, V., Ryll, L., Schölkopf, B., and Hernández-Lobato, J. M. Normflows: A PyTorch Package for Normalizing Flows. *arXiv preprint arXiv:2302.12014*, 2023.

Sutskever, I. *Training Recurrent Neural Networks*. University of Toronto Toronto, ON, Canada, 2013.

Vernon, I., Owen, J., Aylett-Bullock, J., Cuesta-Lazaro, C., Frawley, J., Quera-Bofarull, A., Sedgewick, A., Shi, D., Truong, H., Turner, M., Walker, J., Caulfield, T., Fong, K., and Krauss, F. Bayesian emulation and history matching of JUNE. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2233):20220039, October 2022. doi: 10.1098/rsta.2022.0039.

A. Agent-based models

Agent-based modelling is the name given to a broad approach to modelling complex systems that consist of multiple discrete, autonomous, and heterogeneous interacting components – the "agents" of the system. Examples of such complex systems include the housing market (Baptista et al., 2016), in which a large collection of renters, homeowners, financial institutions etc. interact and take actions which affect, for example, the availability of housing and mortgage rates. An agent-based approach to modelling such a system would model the system at the level of these individual agents in the system, often with the intention of observing how aggregate, macroscopic properties of the system emerge from the microscopic details of the system.

While this is often a natural approach to modelling systems of this kind, the inherently discrete nature of the model's components and dynamics give rise to difficulties in applying gradient-based optimisation and calibration techniques. We expand on these difficulties in section 2.

B. The Brock & Hommes model

The dynamics of the Brock and Hommes model are often expressed as the following system of coupled equations:

$$\mathbf{x}_t = \frac{1}{R}\left[\sum_{j=1}^{J}\left(g_j\mathbf{x}_{t-1}+b_j\right)n_{j,t}+\sigma\epsilon_t\right],\ \epsilon_t\sim\mathcal{N}(0,1), \tag{11}$$

$$n_{j,t} = \frac{\exp\left(\beta U_{j,t-1}\right)}{\sum_{j'=1}^{J}\exp\left(\beta U_{j',t-1}\right)}, \tag{12}$$

$$U_{j,t-1} = \left(\mathbf{x}_{t-1}-R\mathbf{x}_{t-2}\right)\left(g_j\mathbf{x}_{t-3}+b_j-R\mathbf{x}_{t-2}\right), \tag{13}$$

where R, β, σ are auxiliary parameters. We fix $J=4, R=1.01, \sigma=0.04, g_1=b_1=b_4=0, g_4=1.01$, and $\beta=120$ for the experiment presented in the main body of the paper. By rewriting the above system of equations, we are able to find the transition density for observation $\mathbf{x}_{t+1}$ as

$$p(\mathbf{x}_{t+1}\mid\mathbf{x}_{1:t},\boldsymbol{\theta},\boldsymbol{\alpha}) = \mathcal{N}\left(f(\mathbf{x}_{t-2:t},\boldsymbol{\theta},\boldsymbol{\alpha}),\sigma^2/R^2\right) \tag{14}$$

where

$$f\left(\mathbf{x}_{t-2:t},\boldsymbol{\theta},\boldsymbol{\alpha}\right) = \frac{1}{R}\sum_{j=1}^{J}\frac{\exp\left[\beta\left(\mathbf{x}_t-R\mathbf{x}_{t-1}\right)\left(g_j\mathbf{x}_{t-2}+b_j-R\mathbf{x}_{t-1}\right)\right]}{\sum_{j'=1}^{J}\exp\left[\beta\left(\mathbf{x}_t-R\mathbf{x}_{t-1}\right)\left(g_{j'}\mathbf{x}_{t-2}+b_{j'}-R\mathbf{x}_{t-1}\right)\right]}\left(g_j\mathbf{x}_t+b_j\right) \tag{15}$$

and $\boldsymbol{\alpha}=(R,\beta,\sigma)$. The model is taken to be initialised with $\mathbf{x}_{-2}=\mathbf{x}_{-1}=\mathbf{x}_0=0$.

B.1. The asset prices as differentiable and deterministic transformations of input noise

We claim in the main body that we may rewrite the $\mathbf{x}_t$ as deterministic transformations of standard Normal random variables. By exploiting the autoregressive structure of the model, we explicitly provide these forms for $\mathbf{x}_1$ and $\mathbf{x}_2$ below to demonstrate this claim. Throughout, $\epsilon_t\sim\mathcal{N}(0,1)$ are *iid* standard Normal random variables. They are as follows:

$$\mathbf{x}_1 = \frac{1}{R}\left(\sum_{j=1}^{J}\frac{b_j}{J}+\sigma\epsilon_1\right) := f_1(\epsilon_1,\boldsymbol{\theta},\boldsymbol{\alpha}) \tag{16}$$

$$\mathbf{x}_2 = \frac{1}{R}\sum_{j=1}^{J}\frac{\exp\left[\frac{\beta b_j}{R}\left(\sum_{j''=1}^{J}\frac{b_{j''}}{J}+\sigma\epsilon_1\right)\right]}{\sum_{j'=1}^{J}\exp\left[\frac{\beta b_{j'}}{R}\left(\sum_{j''=1}^{J}\frac{b_{j''}}{J}+\sigma\epsilon_1\right)\right]}+\frac{\sigma}{R}\epsilon_2 := f_2(\epsilon_{1:2},\boldsymbol{\theta},\boldsymbol{\alpha}). \tag{17}$$

Repeating this process, we find that the $\mathbf{x}_t$ may all be expressed in the form $\mathbf{x}_t = f_t(\epsilon_{1:t},\boldsymbol{\theta},\boldsymbol{\alpha})$ for a deterministic mapping $f_t:\mathbb{R}^t\times\mathbb{R}^{2J}\times\mathbb{R}^3\to\mathbb{R}$.

Taking

$$\ell(\mathbf{y}, \boldsymbol{\theta}) = \mathrm{MMD}^2(\mathbb{P}_T, \mathbb{P}_{\boldsymbol{\theta}}) \tag{18}$$

$$= \mathbb{E}_{x,x' \sim \mathbb{P}_{\boldsymbol{\theta}}}\left[k(x, x')\right] + \mathbb{E}_{y,y' \sim \mathbb{P}_T}\left[k(y, y')\right] - 2\mathbb{E}_{x \sim \mathbb{P}_{\boldsymbol{\theta}}, y \sim \mathbb{P}_T}\left[k(x, y)\right] \tag{19}$$

$$\approx \frac{1}{T(T-1)} \sum_{t \neq t'} k(\mathbf{x}_t, \mathbf{x}_{t'}) + \frac{1}{T(T-1)} \sum_{t \neq t'} k(\mathbf{y}_t, \mathbf{y}_{t'}) - \frac{2}{T^2} \sum_{t,t'=1}^{T} k(\mathbf{x}_t, \mathbf{y}_{t'}) \tag{20}$$

with a Gaussian kernel k, the loss $\ell(\mathbf{y}, \boldsymbol{\theta})$ is a deterministic and differentiable transformation of the noise drawn from the base distribution ρ of the normalising flow and of the separate noise source with distribution ν given as input to the simulator. This permits us to estimate the gradient of the first term in (5) as

$$\nabla_\phi \mathbb{E}_{q_\phi}\left[\ell(\mathbf{y}, \boldsymbol{\theta})\right] = \nabla_\phi \mathbb{E}_{u \sim \rho}\left[\ell(\mathbf{y}, \boldsymbol{\theta}_\phi(u))\right] \tag{21}$$

$$= \mathbb{E}_{u \sim \rho}\left[\nabla_\phi \ell(\mathbf{y}, \boldsymbol{\theta}_\phi(u))\right] \tag{22}$$

$$= \mathbb{E}_{u \sim \rho}\left[\frac{1}{T(T-1)} \sum_{t \neq t'} \nabla_\phi k\left(f_t(\epsilon_{1:t}, \boldsymbol{\theta}_\phi(u), \boldsymbol{\alpha}), f_{t'}(\epsilon_{1:t'}, \boldsymbol{\theta}_\phi(u), \boldsymbol{\alpha})\right) - \frac{2}{T^2} \sum_{t,t'=1}^{T} \nabla_\phi k(f_t(\epsilon_{1:t}, \boldsymbol{\theta}_\phi(u), \boldsymbol{\alpha}), \mathbf{y}_{t'})\right] \tag{23}$$

where in the first line we use the Law of the Unconscious Statistician and assume throughout that the order of derivatives and integrals can be exchanged freely.

B.2. Further experimental results for the Brock & Hommes model

B.2.1. Calibration results with gradient horizon $H = 0$

In Figure 5, we show the generalised posterior approximated by the converged normalising flow and with gradient horizon $H = 0$, which achieved a loss close to 0. Since the objective function is lower-bounded by 0, this posterior can be taken to be a good approximation to the generalised posterior it targets.

B.2.2. Further evidence in support of the bias-variance trade-off in the reparameterised Monte Carlo gradient estimator at different gradient horizons

To further test the hypothesis that the Monte Carlo gradient estimators at different gradient horizons result in a bias-variance trade-off that can result in favourable performance at finite gradient horizons, we inspect the empirical distribution of the estimates η_N of the gradient (23) based on N Monte Carlo samples,

$$\eta_N := \frac{1}{N} \sum_{n=1}^{N} \left[\frac{1}{T(T-1)} \sum_{t \neq t'} \nabla_\phi k\left(\mathbf{x}_t, \mathbf{x}_{t'}\right) - \frac{2}{T^2} \sum_{t,t'=1}^{T} \nabla_\phi k\left(\mathbf{x}_t, \mathbf{y}_{t'}\right)\right] \tag{24}$$

where here we take a diagonal Gaussian distribution over $\mathbb{R}^4$ as the posterior estimator q_ϕ. In this experiment, therefore, $\phi = (\mu_1, \ldots, \mu_4, \sigma_1, \ldots, \sigma_4)$, where μ_i and σ_i are the mean and standard deviation in each dimension of this choice of q_ϕ.

When implemented in the form given by Equations (14) and (15) – as is necessary to avoid the cumbersome task of finding the explicit form of $f_t(\epsilon_{1:t}, \boldsymbol{\theta}, \boldsymbol{\alpha})$ for each t – the $\mathbf{x}_t$ depend on $\boldsymbol{\theta}$ both explicitly and implicitly via $\mathbf{x}_{t-3:t-1}$. Thus in general we have

$$\frac{\partial \mathbf{x}_t}{\partial \boldsymbol{\theta}_i} = \sum_{l=1}^{\infty} \sum_{\substack{\text{Length } l \text{ paths} \\ \mathbf{v}=(v_0, v_1, \ldots, v_{l-1}, v_l) \\ \text{with } v_0 = \mathbf{x}_t, v_l = \boldsymbol{\theta}_i}} \prod_{i=0}^{l-1} \frac{\partial v_i}{\partial v_{i+1}}, \tag{25}$$

where we abuse notation by taking the derivative on the left-hand side to mean "holding only $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, \ldots, \boldsymbol{\theta}_d)$ constant" while the partial derivatives on the right-hand side are partial derivatives in the true sense of the term (or

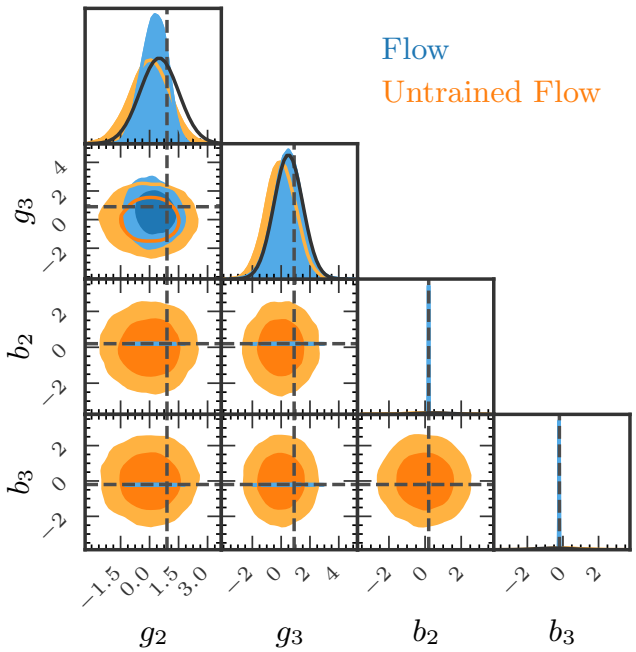

Figure 5. The inferred generalised posterior for the Brock & Hommes model using a gradient horizon $H = 0$.

equivalently the $\mathbf{x}_t$ on the left-hand side is viewed only as a function of $\boldsymbol{\theta}$, while on the right-hand side they are viewed as functions of both $\boldsymbol{\theta}$ and $\mathbf{x}_{1:t}$). Choosing a gradient horizon of $H > 0$ then amounts to retaining paths of length greater than 1 if their first edge corresponds to an edge connecting $\mathbf{x}_t$ to any node in $\mathcal{X}_H := \{\mathbf{x}_{t-h} : h \in \mathcal{H}_H\}$, where $\mathcal{H}_0 = \emptyset$ and $\mathcal{H}_j = \{1, \dots, j\}$ when $j > 0$. In this way, the derivative (25) is then taken instead as

$$\frac{\partial \mathbf{x}_t}{\partial \boldsymbol{\theta}_i} = \frac{\partial \mathbf{x}_t}{\partial \boldsymbol{\theta}_i} + \sum_{l=2}^{\infty} \sum_{\substack{\text{Length } l \text{ paths} \\ \mathbf{v}=(v_0,v_1,\dots,v_{l-1},v_l) \\ \text{with } v_0=\mathbf{x}_t, v_1 \in \mathcal{X}_H, v_l=\boldsymbol{\theta}_i}} \prod_{i=0}^{l-1} \frac{\partial v_i}{\partial v_{i+1}}, \tag{26}$$

where the same abuse of notation is once again used. This elimination of terms from the summation can be expected to reduce the variance since for two random variables X_0, X_1, it is the case that

$$\text{Var}(X_0 + X_1) = \text{Var}(X_0) + \text{Var}(X_1) + 2\text{Cov}(X_0, X_1), \tag{27}$$

which can be greater than $\text{Var}(X_0)$ if $\text{Var}(X_1) + 2\text{Cov}(X_0, X_1) > 0$. It may also be preferable to stricter truncation of the computation graph – for example, by pruning all paths beyond a certain length – as it retains information on long-range dependencies while still potentially reducing variance.

We plot in Figure 6 boxplots for the distribution of η_N obtained with $N = 5$ and with different values of H, obtained at a fixed value for ϕ (the results were qualitatively similar for different ϕ we tried, and so we show only the results from one settings). We also show the same boxplots for the gradient estimate obtained with the score-based estimator,

$$\nabla_\phi \mathbb{E}_{q_\phi}\left[\ell(\mathbf{y}, \boldsymbol{\theta})\right] = \mathbb{E}_{q_\phi}\left[\ell(\mathbf{y}, \boldsymbol{\theta}) \nabla_\phi \log q_\phi(\boldsymbol{\theta})\right] \approx \frac{1}{N} \sum_{n=1}^{N} \ell(\mathbf{y}, \boldsymbol{\theta}^{(n)}) \nabla_\phi \log q_\phi(\boldsymbol{\theta}^{(n)}). \tag{28}$$

Orange (green) dashed lines show the mean (median) of the distributions. The blue crosses show the mean of the distribution of the gradient estimate (28) obtained using $N = 1000$, which provide a good estimate of the target value (since the score-based estimator is unbiased). We see from this that, generally speaking, the variance of these estimates increase as H increases, while the bias in the estimates do not degrade substantially. This highlights the possibility that using a

finite gradient horizon can be beneficial when performing Monte Carlo gradient estimation for differentiable time series simulation models, such as ABMs, when reparameterisation is possible. Further work will be required to establish the general applicability and suitability of this technique.

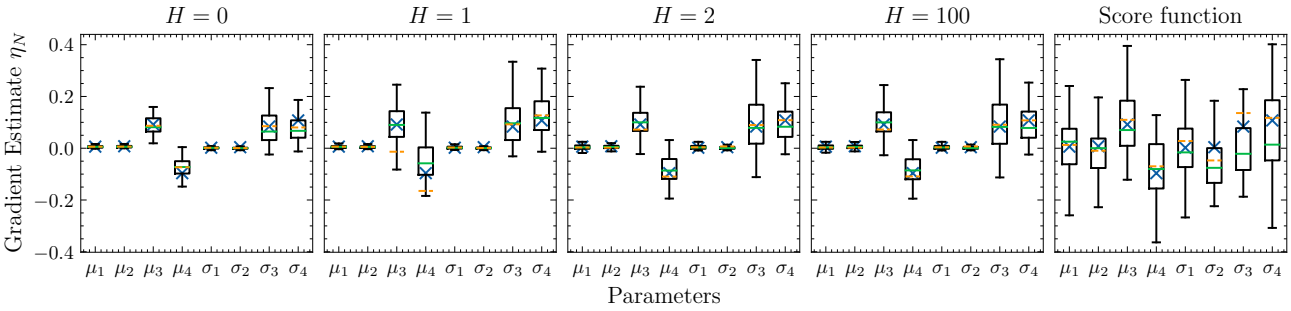

Figure 6. Boxplots for the empirical distribution of Monte Carlo gradient estimates from gradient horizons $H = 0, 1, 2, 100$, and from the score-based gradient estimate, (28). Orange (green) dashed lines show the mean (median) of the distributions. Blue crosses show the mean of the distribution of the gradient estimate (28) obtained using $N = 1000$.

C. The JUNE model

The JUNE model (Aylett-Bullock et al., 2020) is an agent-based epidemiological model that generates a synthetic population at a highly detailed level using the English census data. This model has been applied in various scenarios, including analyzing the impact of the first and second waves of SARS-CoV2 in England (Vernon et al., 2022) and devising strategies to control disease transmission in refugee settlements (Aylett-Bullock et al., 2021).

To enhance its performance and enable gradient-based calibration, the JUNE model has been incorporated into the GRADABM framework (Chopra et al., 2023; Quera-Bofarull et al., 2023). This integration allows for faster execution and more efficient parameter calibration. The JUNE model offers a wide range of configurable parameters related to disease transmission and progression, vaccination, and non-pharmaceutical interventions.

Given a susceptible agent exposed to an infection at location L, the probability of that agent getting infected is given b

$$p = 1 - \exp\left(-\psi_s \, \beta_L \, \Delta t \sum_{i \in g} \mathcal{I}_i(t)\right), \tag{29}$$

where the summation is conducted over all contacts an agent has with infected individuals at the given location L. The term $\mathcal{I}_i(t)$ represents the time-dependent infectious profile of each infected agent, while Δt is the duration of the interaction. Additionally, β_L corresponds to a location-specific parameter that captures the variation in the nature of interactions across different locations.

Since the β_L parameters are not directly measurable physical quantities, they are typically calibrated using available data on the number of cases or fatalities over a specific time period. For the current work we consider the calibration of 11 β_L parameters corresponding to the contact intensity at households, companies, schools, universities, pubs and restaurants, gyms, cinemas, shops, care homes, and residence visits. Additionally, we also calibrate the initial number of infections, I_0, which are distributed randomly across the population. The synthetic ground truth data is generated by using $I_0 = 10^{-3.5} N_a$ where N_a is the number of agents, and $\beta_{\text{household}} = \beta_{\text{carehome}} = 0.6$, $\beta_{\text{school}} = \beta_{\text{company}} = \beta_{\text{university}} = 0.4$, $\beta_{\text{pub}} = \beta_{\text{shop}} = \beta_{\text{gym}} = \beta_{\text{cinema}} = \beta_{\text{visit}} = 0.1$.

The normalising flow is trained setting $\ell(\mathbf{y}, \boldsymbol{\theta})$ to be the squared distance between the $\log_{10}$ of the infection time-series. This choice of loss function keeps the training robust against outliers, since the number of infections can oscillate between several orders of magnitude. The specific training parameters are described in Appendix D.

C.1. Further experimental results for the JUNE model

We show in Figure 7 the loss as a function of epoch when performing SVI with GRADABM-JUNE. We observe a rapid convergence after 600 epochs. Since we are sampling 5 Monte Carlo samples to estimate Equation 3, this results in 3,000

model evaluations. We also make a corner plot of the train and untrained normalising flow which we show in Figure 8, where the solid black line denotes the prior density. We observe that the flow is very confident about the value of more sensitive parameters such as the initial number of infections and the contact intensity at companies, while it is less certain for venues which have a low impact in the overall number of infections, such as cinemas. It is worth noting that this calibration challenge is very underdetermined, that is, it is quite difficult by just observing the overall number of infections over time to infer the contact intensities at each location. Nonetheless, the flow fits well the synthetic ground truth data.

D. Further experimental details

We use the NORMALIZING-FLOWS library (Stimper et al., 2023) to implement the normalising flows in PyTorch. All models are trained using the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 10^{-3}.

To calibrate the Brock & Hommes and JUNE models, we employ a masked affine autoregressive flow (Papamakarios et al., 2017) with 16 transformations, each parametrized by 2 blocks with 20 hidden units. We also set the regularisation weight to $w = 10^{-3}$ for both models and estimate Equation 3 using 5 Monte Carlo samples.

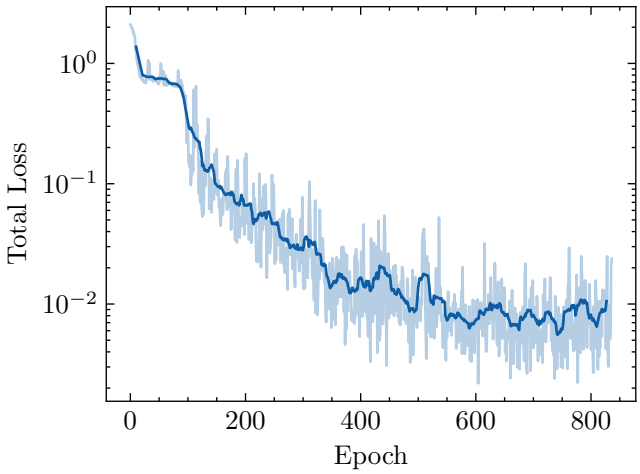

Figure 7. Loss function per epoch for the JUNE calibration.

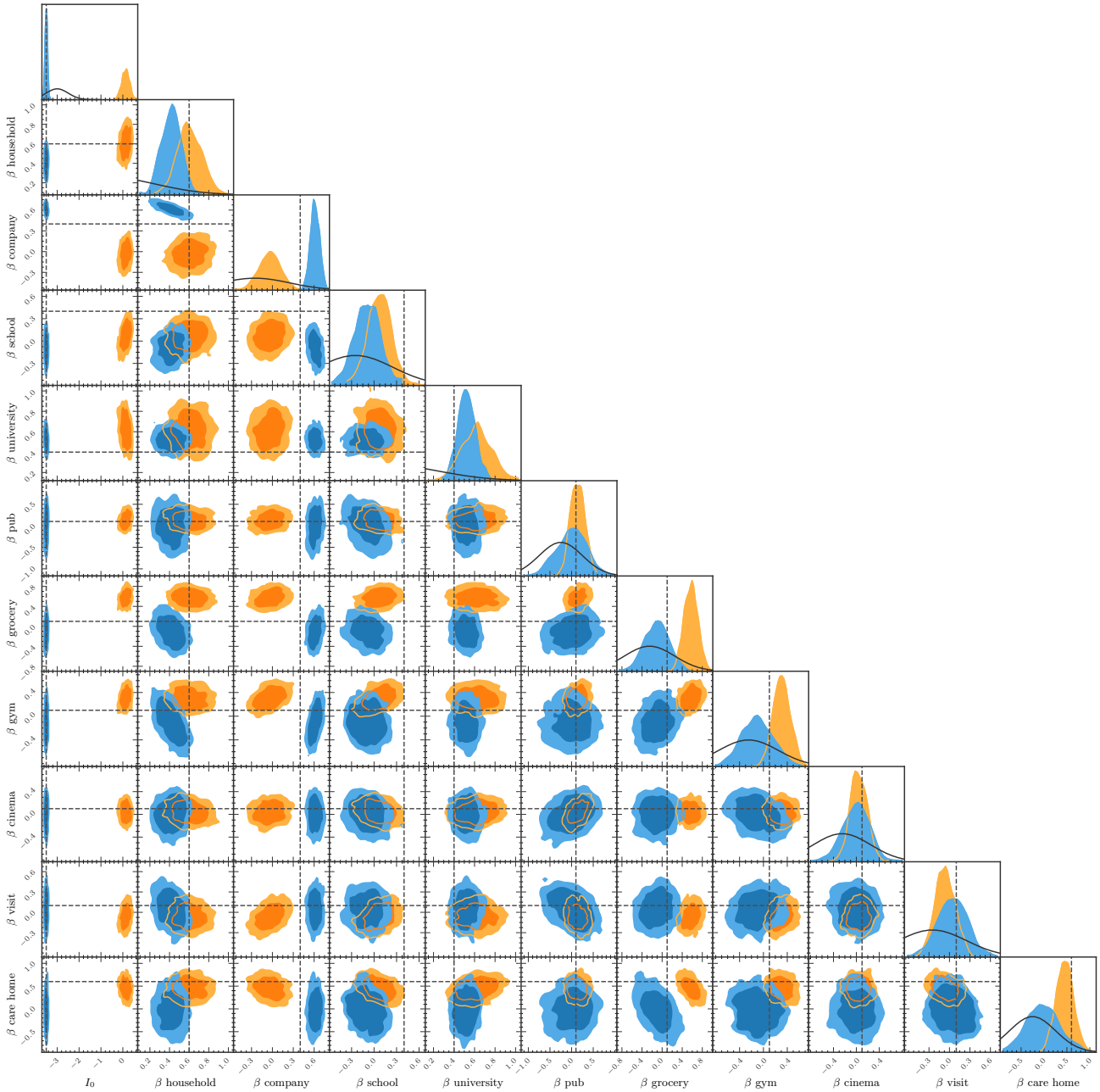

Figure 8. Corner plot of samples from the untrained normalising flow, the trained normalising flow, and the prior for the JUNE model calibration.