Bamojax: Bayesian Modelling with JAX

Max Hinne ⬤ 1*

1 Radboud University, Nijmegen, The Netherlands * These authors contributed equally.

Summary

Bayesian statistics offers a principled and elegant framework for inferring hidden causes from observed effects. It also provides a rigorous approach to hypothesis testing (model comparison), with advantages such as built-in complexity penalties, and the ability to quantify evidence in favour of the null hypothesis.

However, exact Bayesian inference is computationally intractable in all but the simplest of cases, and requires *approximate inference* techniques, such as Markov chain Monte Carlo (MCMC) and variational inference. Recent advances in the Python JAX (Bradbury et al., 2018) framework have enabled highly efficient implementations of these algorithms, due to features such as automated differentation and GPU acceleration. These developments have the potential to greatly increase the efficiency of statistical modelling pipelines.

bamojax ('Bayesian Modelling in Jax') is a probabilistic programming language (PPL) that combines ease-of-use with access to advanced inference algorithms implemented in the Jax ecosystem.

Statement of need

Bamojax is a Bayesian modelling tool based on Python & JAX (Bradbury et al., 2018). It provides an intuitive, intermediate-level interface between defining a Bayesian statistical model conceptually, and performing efficient inference using the BlackJAX package (Cabezas et al., 2024).

Existing probabilistic programming languages, such as PyMC (Abril-Pla O, 2023), can export (the logarithm of) a probability density function that enables BlackJAX-based inference. However, this has two limitations:

1. It does not support Gibbs sampling, where variables are updated individually using their own MCMC kernels. For example, when approximating the posterior over a latent Gaussian process (GP) and its hyperparameters, elliptical slice sampling (Murray et al., 2010) for the GP can be more efficient than applying No-U-Turn Hamiltonian Monte Carlo (NUTS HMC; (Hoffman & Gelman, 2014)) sampling to all variables jointly. This becomes even more important when embedding MCMC sampling in Sequential Monte Carlo (Hinne, 2025).
2. It makes it harder to apply tempered Sequential Monte Carlo methods that need separate prior and likelihood terms.

While users can circumvent these issues by manually implementing their models using BlackJAX, this is a labor-intensive and error-prone process. **Bamojax** addresses this gap by providing a user-friendly interface for model construction and Gibbs sampling on top of BlackJAX.

In **Bamojax**, users can define a probabilistic model by specifying variables as well as their associated distributions and dependencies, structured using a directed acyclic graph (DAG).

40 Under the hood, **Bamojax** translates this DAG and collection of probability distributions to the
41 probability densities used in the approximate inference, leveraging the probability definitions
42 defined in NumPyro (Phan et al., 2019). This abstraction allows users to focus on the
43 conceptual model formulation, rather than the mathematical or inference details, leading to a
44 more intuitive, less error-prone, and more efficient development workflow.

45 **Bamojax** is designed for researchers, students, and practitioners that want to make use of
46 the extremely fast approximate inference offered by BlackJAX, but want to focus on model
47 development instead of implementation.

Comparison with existing tools

49 **Bamojax** is comparable to existing modern probabilistic programming languages, like NumPyro
50 (Phan et al., 2019), Oryx ((jax-ml), 2025), and PyMC (Abril-Pla O, 2023), that use the fast
51 JAX backend for efficient computation of Bayesian inference. The aim of **Bamojax** is to stay
52 close to (JAX) Numpy levels of user control in defining probability densities and transformations,
53 while at the same time providing some level of abstraction to the user, by automatically deriving
54 densities from NumPyro primitives and the directed acylcic graph structure of a Bayesian
55 model. This allows for convenient integration of new methods with existing tools.

56 **Bamojax** easily admits Gibbs sampling, where individual model parameters are updated in
57 turn, which in practice can lead to efficiency gains and sampling of discrete variables. These
58 benefits increase further when Gibbs sampling is used with Sequential Monte Carlo (Hinne,
59 2025), which is straightforward to set up here.

60 Furthermore, **Bamojax** provides a convenient interface to the different sampling algorithms
61 that are available in BlackJAX, giving the user fine-grained control over the inference strategy
62 of their models. This enables users to mix-and-match BlackJAX MCMC kernels with elements
63 of their probabilistic model, while maintaining the efficiency of JAX-based inference.

Acknowledgements

References

67 Abril-Pla O, C. C., Andreani V. (2023). PyMC: A modern and comprehensive probabilistic
68 programming framework in Python. *PeerJ Computer Science*, *9*(e1516). https://doi.org/
69 10.7717/peerj-cs.1516

70 Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G.,
71 Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable
72 transformations of Python+NumPy programs* (Version 0.4.35). http://github.com/jax-ml/
73 jax

74 Cabezas, A., Corenflos, A., Lao, J., & Louf, R. (2024). *BlackJAX: Composable Bayesian
75 inference in JAX*. https://arxiv.org/abs/2402.10797

76 Hinne, M. (2025). An introduction to Sequential Monte Carlo for Bayesian inference and model
77 comparison—with examples for psychology and behavioral science. *Behavior Research
78 Methods*, *57*(25). https://doi.org/10.3758/s13428-025-02642-1

79 Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path
80 lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*(47),
81 1593–1623. http://jmlr.org/papers/v15/hoffman14a.html

82 (jax-ml), O. contributors. (2025). *Oryx: Probabilistic programming and deep learning in JAX*
83 (Version 0.2.9). https://github.com/jax-ml/oryx

Murray, I., Adams, R., & MacKay, D. (2010). Elliptical slice sampling. In Y. W. Teh & M. Titterington (Eds.), *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (Vol. 9, pp. 541–548). PMLR. https://proceedings.mlr.press/v9/murray10a.html

Phan, D., Pradhan, N., & Jankowiak, M. (2019). Composable effects for flexible and accelerated probabilistic programming in NumPyro. *arXiv Preprint arXiv:1912.11554*.