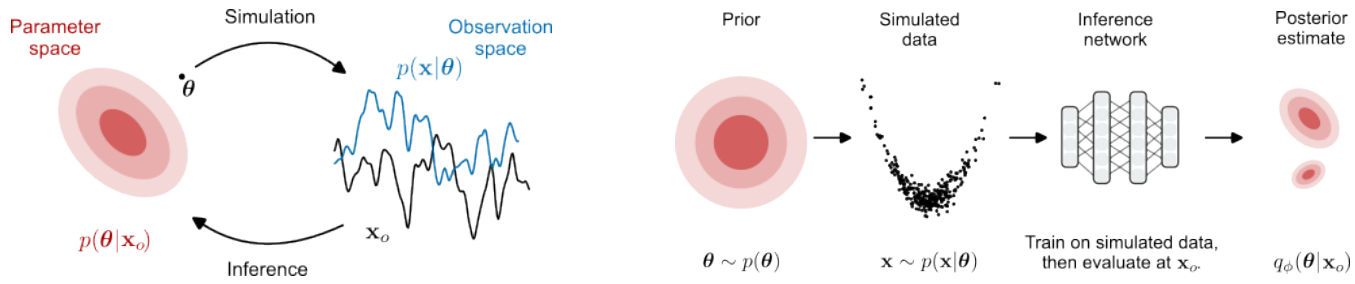
Online Simulation-Based Inference for Large-Scale Scientific Models

"Exploring how diffusion and transformer-based models can infer the laws behind complex scientific simulations"

Supervision:

- Bruno Raffin, DataMove, Inria (bruno.raffin@inria.fr)
- Pedro L. C. Rodrigues, Statify, Inria (pedro.rodrigues@inria.fr)

Context Researchers are turning to machine learning to tackle various problems in science, from biology to astrophysics and fluid dynamics. The project that we propose is part of this growing AI4Science movement, focusing on a key challenge in experiments: figuring out which model parameters best match the data we observe (Figure 1). More specifically, we use **simulation-based inference** (SBI) [1], a Bayesian approach that leverages deep generative models, such as conditional normalizing flows and score-diffusion models, to approximate the posterior distribution – assigning higher probability to parameter values most likely to have produced an observed data.



(a) Simulation-based inference loop

(b) Neural posterior estimation pipeline

Figure 1: Given a stochastic simulator taking parameters θ as input and returning simulations $x \sim p(x|\theta)$, the posterior distribution $p(\theta|x_0)$ helps us determine the parameters which are the most likely to have generated observation x_0. (b) SBI consist of four main steps: (i) draw parameters from the prior distribution $\theta_i \sim p(\theta)$, and (ii) run the simulator to generate data $x_i \sim p(x|\theta_i)$. (iii) Train over dataset (θ_i, x_i) a conditional generative model q_ϕ that takes x as input and predicts a distribution over parameters θ. (iv) Use $q_\phi(\theta|x_0)$ as an approximation to the posterior $p(\theta|x_0)$. Figure taken from [1].

Despite recent successes of the SBI framework across various applied domains, its applicability is currently constrained to relatively small-scale models. **The primary goal of this project is to extend the capabilities of SBI to accommodate simulators that rely on solving large systems of differential equations to generate observations.** As such, the candidate will have the opportunity to work in the exciting intersection between modern machine learning methods (e.g. sampling with diffusion models, embeddings with transformers, training with flow matching) and high performance computing (e.g. handling large-scale parallel simulators, multi-node and GPU training on large supercomputers).

Methods When considering large scale simulations, the amount of data produced can be overwhelming and the execution time too long, calling to resort to supercomputers and High Performance Computing (HPC). To optimize performance and reduce costs (power, storage, compute time), training can be performed online. Multiple simulations are executed concurrently and continuously to produce data that are used asap, without being stored in files, by a training process that also runs concurrently with these simulations (Fig. 2) [2, 3]. The traditional SBI workflow

consisting of (simulate → store, then store → train) has to be re-visited to properly support and leverage this online training workflow (simulate → buffer → train).

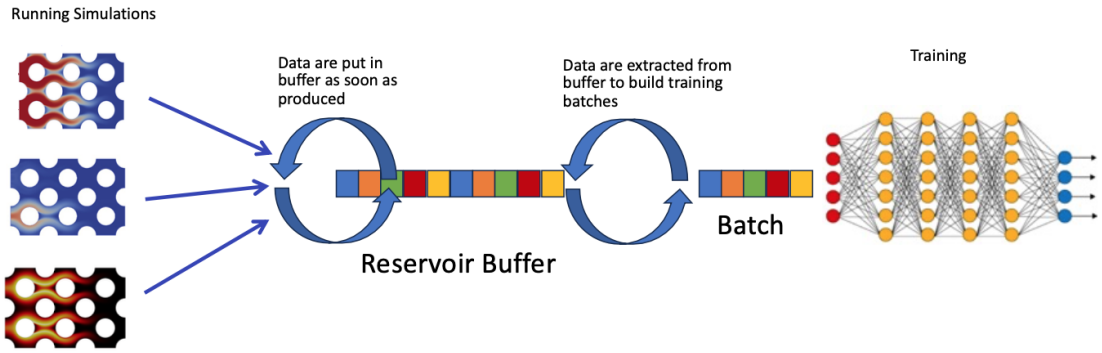


Figure 2: Schematic view of an online training scheme

To be more specific, consider the usual SBI approach of training a conditional neural density estimator q_ϕ that approximates the target posterior through the minimization of

$$\mathcal{L}(\phi) = -\frac{1}{N}\sum_{i=1}^{N} \log q_\phi(\theta_i|\boldsymbol{x}_i) \quad \text{and} \quad (\theta_i, x_i) \sim p(\theta, x) = p(\theta)p(x|\theta)\ . \tag{1}$$

The batch of N pairs of parameters (θ_i) and simulations (x_i) is provided upfront and the loss function is minimized via some variant of stochastic gradient descent. Note that this is well motivated because when $N \to \infty$, one can show that the minimizer of Equation 1 is indeed the target posterior $p(\theta|x)$. However, it is not clear how the minimization behaves when the training samples are obtained sequentially due to simulation constraints and/or sampled from a different distribution than $p(\theta, x)$ as one would do when trying to reduce the number of calls to the simulator.

During the M2 internship, the candidate will explore the following questions:

- What is the direct impact of an online paradigm for simulations on the usual batched SBI training? What are the precise bottlenecks and challenges to this transition?
- Are there other loss functions more appropriate to minimize instead of Equation 1 when working under the paradigm of large-scale simulators?
- Can approaches from online reinforcement learning to make smart queries to the simulator and minimize total cost of compute can be reused and adapted?

N.B.: During the research internship, if the candidate proposes other pertinent questions related to the project, he or she will be free to tackle them under the supervisors' guidance. Moreover, the internship is part of a NumPEx funding with the goal of transitioning to a Ph.D. thesis starting in October 2026.

Environment The candidate will be supervised by Bruno Raffin (Inria Grenoble) and Pedro L. C. Rodrigues (Inria Grenoble). He or she will work mainly at the DataMove team, located at the IMAG building in UGA campus, and in close collaboration with the Statify team, located in Inria Montbonnot. He or she will have access to a team of experts in high-performance computing and machine learning that will help him or her to kickstart the project under the best conditions. The candidate will also have access to supercomputers to run experiments. The position comes with salary in line with current university positions and subsidized meals. For more information, please contact the supervisors.

Requirements

- Strong mathematical background, specially advanced concepts in machine learning and statistics.
- Good working knowledge on Python and its scientific computing ecosystem (`scipy`, `numpy`, `pytorch`, etc).

- Some practical experience with running experiments on parallel machines will be a plus.
- Excellent writing and oral skills in French and English.

References

[1] Michael Deistler, Jan Boelts, Peter Steinbach, Guy Moss, Thomas Moreau, Manuel Gloeckler, Pedro LC Rodrigues, Julia Linhart, Janne K Lappalainen, Benjamin Kurt Miller, et al. Simulation-based inference: A practical guide. *arXiv preprint arXiv:2508.12939*, 2025.

[2] Sofya Dymchenko and Bruno Raffin. Loss-driven sampling within hard-to-learn areas for simulation-based neural network training. In *MLPS 2023-Machine Learning and the Physical Sciences Workshop at NeurIPS 2023-37th conference on Neural Information Processing Systems*, pages 1–5, 2023.

[3] Lucas Thibaut Meyer, Marc Schouler, Robert Alexander Caulk, Alejandro Ribés, and Bruno Raffin. High throughput training of deep surrogates from large ensemble runs. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16, 2023.