Diffusion Posterior Sampling for Simulation-Based Inference in Tall Data Settings

Julia Linhart[1,2] Gabriel Victorino Cardoso[3] Alexandre Gramfort[1,2] Sylvain Le Corff[4] Pedro L. C. Rodrigues[1,5]

[1]Inria [2]Université Paris-Saclay [3]CMAP, École Polytechnique [4]LPSM, Sorbonne Université [5]Université Grenoble Alpes

Background

Simulation-Based Inference. Scientific models are often defined as stochastic simulators:

$$\theta \sim \lambda(\theta) \rightarrow \boxed{\text{Stochastic Simulator}} \rightarrow x \sim p(x|\theta)$$

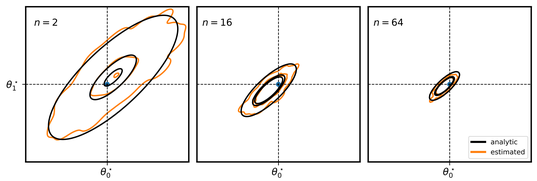
$$p(\theta|x) \propto \lambda(\theta)\, p(x|\theta)$$

$p(x|\theta)$ is intractable, but we can simulate $(\theta_i, x_i) \sim p(\theta, x)$, and train a network *once* to approximate $p(\theta|x^\star)$ for any $x^\star$.

Tall Data Setting. For n i.i.d. observations $x^\star_{1:n} \sim p(x|\theta^\star)$, the tall posterior factorizes as

$$p(\theta \mid x^\star_{1:n}) \propto \lambda(\theta)^{1-n} \prod_{j=1}^{n} p(\theta \mid x^\star_j) \tag{1}$$

and concentrates around $\theta^\star$ as $n \to \infty$ (Bernstein-von Mises).



Related Work

Method	Direct post.	Sim.-effic.	MCMC-free
Aug. NPE + DeepSet *(Radev et al., 2020)*	✓	✗	✓
Factorized NLE/NRE *(Hermans et al., 2020)*	✗	✓	✗
F-NPSE *(Geffner et al., 2023)*	✓	✓	✗
Ours (JAC / GAUSS)	✓	✓	✓

- Simulation-efficient methods require MCMC at inference.
- MCMC-free methods rely on augmented datasets.

Can we compose individual NPSE scores to directly sample the tall posterior — without MCMC?

Method

Setup. Forward diffusion: $q_{t|0}(\theta|\theta_0) = \mathcal{N}(\sqrt{\alpha_t}\theta_0, v_t\mathbf{I})$. Train NPSE via DSM on $(\theta_i, x_i) \sim p(\theta, x)$: $s_\varphi(\theta,\, x,\, t) \approx \nabla_\theta \log p_t(\theta \mid x)$. At inference, $s_\varphi(\theta, x^\star_j, t) =: s_{\varphi,j,t}(\theta)$ gives the score at each $x^\star_j$.

Key result — Tractable approximate diffused tall posterior score

$$\nabla_\theta \log p_t(\theta \mid x^\star_{1:n}) \;\approx\; \Lambda^{-1}\left[\sum_{j=1}^{n} \Sigma^{-1}_{t,j} \underbrace{s_{\varphi,j,t}(\theta)}_{\text{NPSE}} + (1-n)\, \Sigma^{-1}_{t,\lambda} \underbrace{\nabla_\theta \log p^\lambda_t(\theta)}_{\text{prior score}}\right] \tag{2}$$

with $\Lambda = \sum_j \Sigma^{-1}_{t,j} + (1-n)\Sigma^{-1}_{t,\lambda}$ positive definite and $\Sigma_{t,j/\lambda}$ the backward diffusion covariances.

✓ Explicit approx. of the diffusion process – enable DDIM, no MCMC.
✓ Single amortized NPSE — no augmented dataset.

Derivation in 3 steps.

1. *Exact decomposition into individual scores (via (1) & Fisher's identity):*

$$\nabla_\theta \log p_t(\theta \mid x^\star_{1:n}) = \nabla_\theta \log \int q_{t|0}(\theta \mid \theta_0)\, p(\theta_0 \mid x^\star_{1:n})\, \mathrm{d}\theta_0$$

$$= (1-n)\, \nabla_\theta \log p^\lambda_t(\theta) + \sum_{j=1}^{n} \nabla_\theta \log p_t(\theta \mid x^\star_j) + \underbrace{\nabla_\theta \log \mathcal{L}_\lambda(\theta, x^\star_{1:n})}_{\text{intractable}}$$

where $\mathcal{L}_\lambda$ is defined by the backward kernels needed to run DDIM.

2. *Second-order Tweedie approximation of backward kernels:*

$$\hat{q}_{0|t}(\theta_0|\theta, x^\star_j) = \mathcal{N}(\mu_{t,j}(\theta),\ \Sigma_{t,j}(\theta)) \quad \Rightarrow \ \textbf{(2)}+F$$
$$\hat{q}^\lambda_{0|t}(\theta_0|\theta) = \mathcal{N}(\mu_{t,\lambda}(\theta),\ \Sigma_{t,\lambda}(\theta))$$

with $\mu_{t,j}(\theta)$, $\Sigma_{t,j}(\theta)$ in closed form *(Boys et al., 2023)*, making **(2)** tractable up to a residual term F.

3. *Constant covariance assumption:*

$$\nabla_\theta \Sigma_{t,j}(\theta) = \nabla_\theta \Sigma_{t,\lambda}(\theta) = 0 \ \Rightarrow \ F = 0 \ \Rightarrow \ \textbf{Key result (2)}$$

Algorithm

Input: NPSE s_φ, observations $x^\star_{1:n}$
Sample $\theta_T \sim \mathcal{N}(0, \mathbf{I})$
for $t = T, \ldots, 1$**:**
1. Get scores $s_{\varphi,j,t}(\theta_t)$; prior score
2. Estimate $\hat{\Sigma}_{t,j}$: **JAC** or **GAUSS**
3. Compute $s^\star$ via **(2)**
4. DDIM: $\theta_{t-1} \leftarrow \mathrm{DDIM}(\theta_t, s^\star)$

return $\theta_0 \sim p(\theta \mid x^\star_{1:n})$

Strategies for $\hat{\Sigma}_{t,j}$

JAC — theoretically exact

$$\hat{\Sigma}^{-1}_{t,j} = \frac{\alpha_t}{v_t}(\mathbf{I} + v_t\, \nabla_\theta s_{\varphi,j,t})^{-1}$$

via Jacobian of NN — expensive, unstable · constant via stop-grad

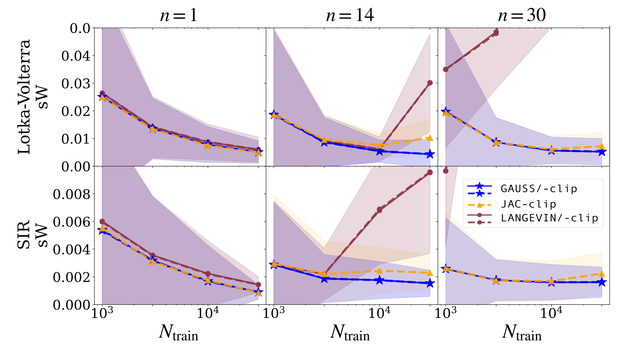
GAUSS — closed-form approximation

$$\hat{\Sigma}^{-1}_{t,j} = \hat{\Sigma}^{-1}_{0,j} + \frac{\alpha_t}{v_t}\mathbf{I}$$

emp. cov. of 100-step DDIM samples per obs. · constant by definition

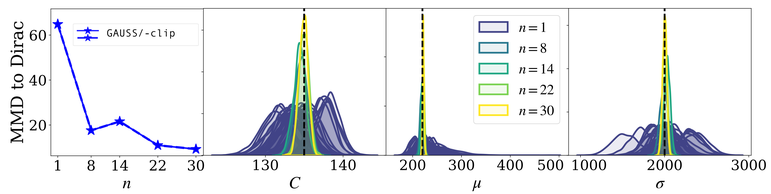
Results

SBI Benchmarks. SIR and Lotka-Volterra. Metric: Sliced Wasserstein vs. MCMC reference as a function of N_{train} and for increasing $n \in \{1, 14, 30\}$.



✗ **LANGEVIN** diverges for $n \geq 14$ ✓ **GAUSS** converges for all n ✓ **JAC**-clip competitive

Real-world application: Jansen-Rit Neural Mass Model. Stochastic simulator of EEG-like signals, with parameters $\theta = (C, \mu, \sigma, g)$ and fixed $g = 0$, $N_{\text{train}} = 50\,000$.



✓ MMD decreases with n ✓ Marginals concentrate around true $\theta^\star$

indeterminacy for (μ, σ) in 4D ⇒ coupling with g ⇒ hierarchical model *(Rodrigues et al., 2021)*

Conclusion

✓ Tractable tall posterior score — one network, no MCMC, no augmented dataset
✓ **GAUSS**: more robust, accurate and ∼2.5× faster than F-NPSE
✓ Validated on synthetic benchmarks and a real-world neuroscience simulator

Extensions & Perspectives.

- Partial factorization to mitigate error acc. for large n *(Appendix)*
- **JAC** stabilization strategies *(Gloeckler et al., 2025)*
- Extension to large-scale hierarchical models *(Arruda et al., 2025)*
- Accurate and efficient model criticism...? *(ongoing work)*

Link to paper!

julia.linhart@nyu.edu github.com/JuliaLinhart

TMLR, J2C Award, ICML 2026 | arXiv:2404.07593