Fast and Robust Simulation-Based Inference With Optimization Monte Carlo

Vasilis Gkolemis[1,2] Christos Diou[2] Michael U. Gutmann[3]

[1]ATHENA Research Center [2]Harokopio University of Athens [3]University of Edinburgh

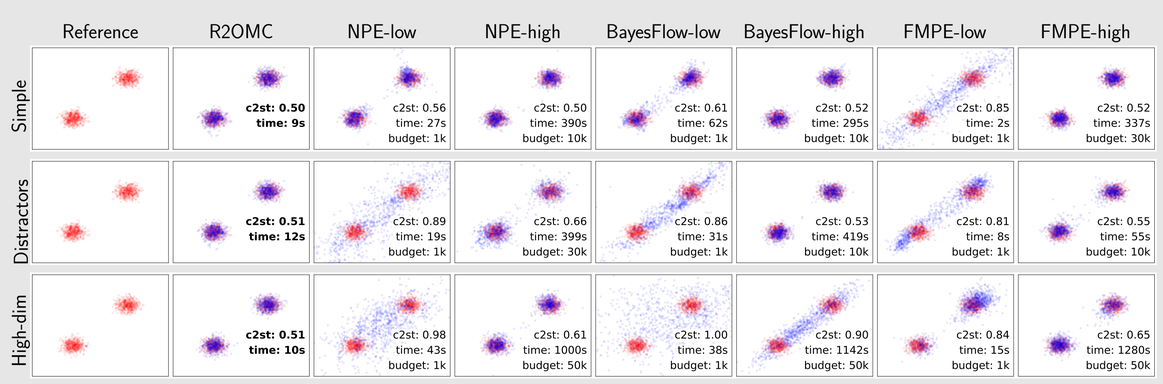
R2OMC: In a nutshell

SBI with a *differentiable* simulator?

⇒ R2OMC infers accurate posterior samples efficiently

Speed vs. Accuracy vs. Transparency in SBI

Traditional LFI methods (ABC, SMC-ABC, MCMC-ABC) are transparent but do not scale beyond low-dimensional parameter spaces.
Modern neural methods (NPE, BayesFlow, FMPE) scale to high dimensions but require large simulation budgets, long runtimes, and sacrifice transparency.
R2OMC is fast (low simulation budget), scalable (high-dimensional spaces), and transparent (no neural network).

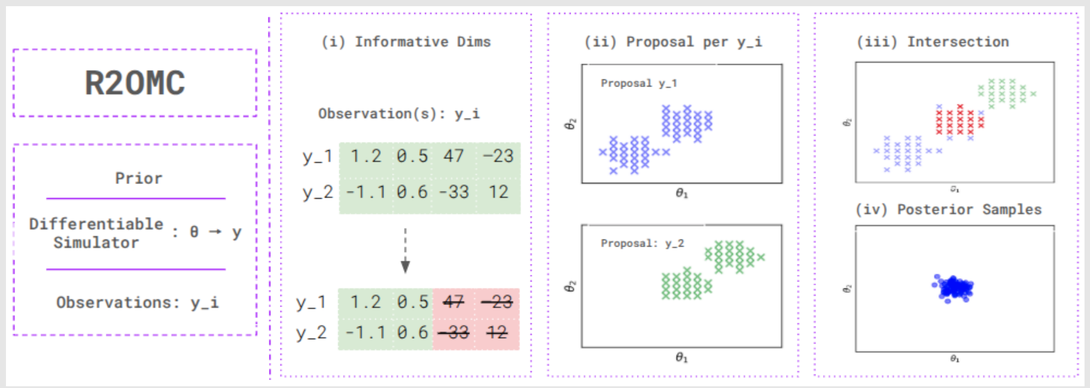


Two-Gaussian mixture: simple ($D=D_y=2$), distractors ($D=2, D_y=18$), high-dim ($D=D_y=10$). R2OMC matches the reference at low runtime; neural baselines need much higher budgets.

R2OMC builds on top of OMC and ROMC

- **OMC** (Meeds et al., 2015) and **ROMC** (Ikonomov et al., 2020):
  - fix random seeds $\mathbf{u}_i$ and turn inference into S deterministic problems
  - ABC likelihood: $L_\epsilon(\boldsymbol{\theta}) = \Pr(d(\mathbf{y}, \mathbf{y}^o) \leq \epsilon \mid \boldsymbol{\theta})$.
  - $\boldsymbol{\theta}_i^* = \arg\min_{\boldsymbol{\theta}} \; d(g(\boldsymbol{\theta}, \mathbf{u}_i), \; \mathbf{y}^o)$; samples drawn from proposals q_i around each optimum.
- **Limitations of vanilla ROMC:**
  - No strategy for **distractors** – inflated ϵ, broad posteriors.
  - No strategy for **multiple iid observations**.
  - No optimized implementation – can only tackle low-dim problems.

R2OMC: Method overview



Gradient-based optimization.
Distances $d_n(\boldsymbol{\theta}) = \|g(\boldsymbol{\theta}, \mathbf{u}_i) - \mathbf{y}_n^o\|$ minimized via gradient descent; scales to high-dimensional $\boldsymbol{\theta}$.

Distractor filtering.
Average gradient norm per output dimension thresholded at τ:
$I_i = \mathbb{E}_{p(\boldsymbol{\theta})p(\mathbf{u})}[\, \|\nabla_{\boldsymbol{\theta}} g_i(\boldsymbol{\theta}, \mathbf{u})\| \,] > \tau$. Optimization runs only on informative dimensions $\mathbf{m}_\tau$.

Multiple iid observations.
Per-observation proposals combined: $q(\boldsymbol{\theta}) = \frac{1}{NS} \sum_{n,i} q_i^n(\boldsymbol{\theta})$, with weights $w_p = \frac{p(\boldsymbol{\theta}_p)}{q(\boldsymbol{\theta}_p)} \prod_n \sum_i \mathbf{1}_{C_\epsilon^{i,n}}(\boldsymbol{\theta}_p)$ retaining samples consistent with *all* observations.

JAX implementation.
`grad` for Jacobians, `vmap` vectorizes over seeds and dimensions, `jit` fuses optimization steps.

How to use R2OMC

Install: `pip install git+https://github.com/givasile/lfi`

```python
from lfi import priors, simulators, inference

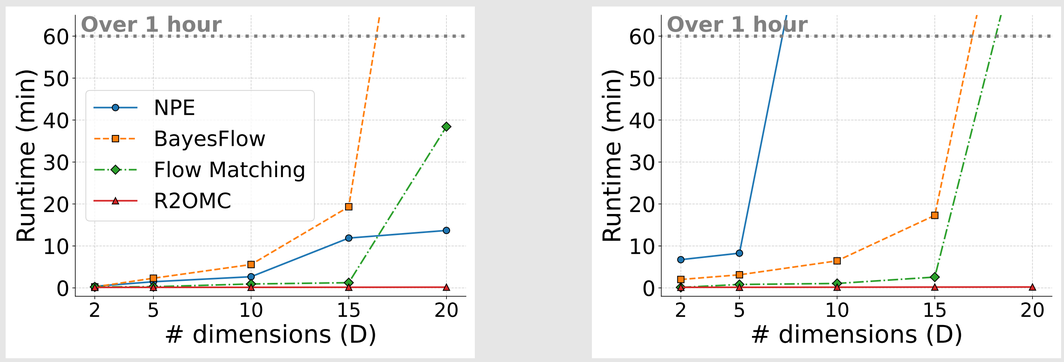
prior     = priors.UniformPrior(low=-3, high=3, dim=2) # or your own prior
simulator = simulators.TwoMoons(dim=2, dim_y=2)        # or your own simulator
y_obs     = ...                                        # shape (N_obs, D_y)

model = inference.r2omc.R2OMC(prior, simulator, y_obs)
model.fit(budget=500)
samples = model.sample(nof_samples=1000)
```

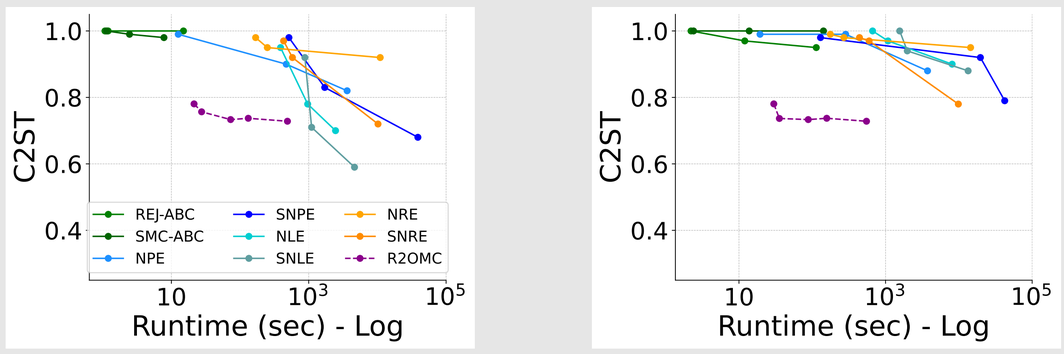
R2OMC: Paper & Code

- **Paper:** arXiv:2504.11087
- **Code:** github.com/givasile/lfi

Experiments: Gaussian & SLCP benchmarks

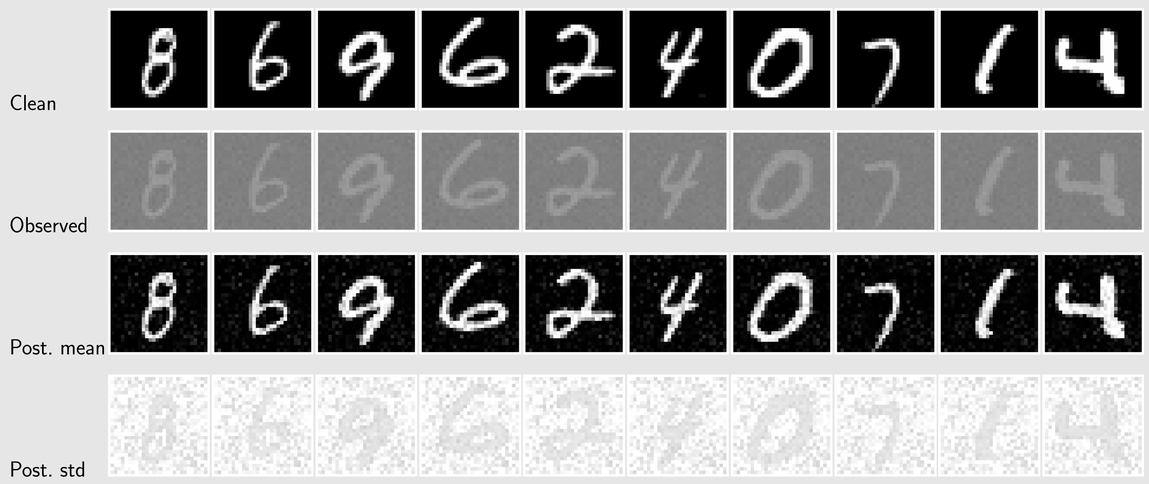


Success Frontier on S_{base} (left) and $S_{\text{base}}^{\text{dist}}$ (right): minimum runtime to reach C2ST ≤ 0.75. R2OMC stays in seconds; neural-based methods blow up past one hour.



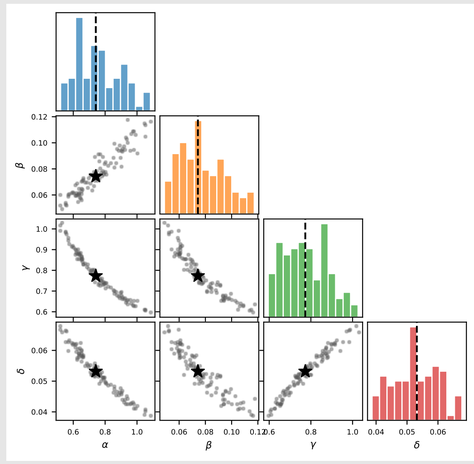
C2ST vs. runtime on SLCP (left) and SLCP+distractors (right). R2OMC reaches competitive accuracy in seconds; neural-based methods require high runtimes.

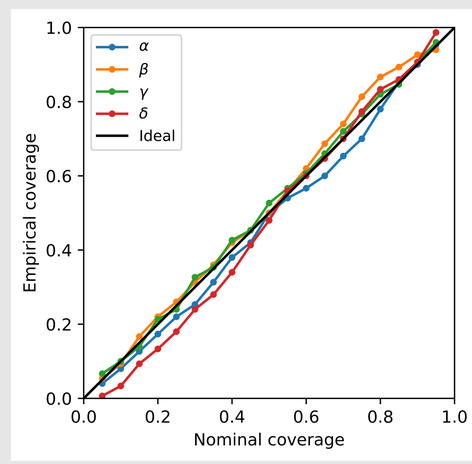
Experiments: 784D image inference

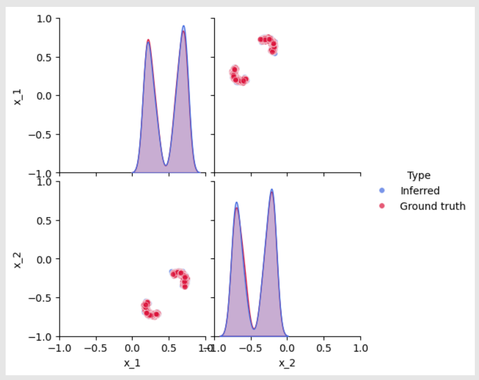


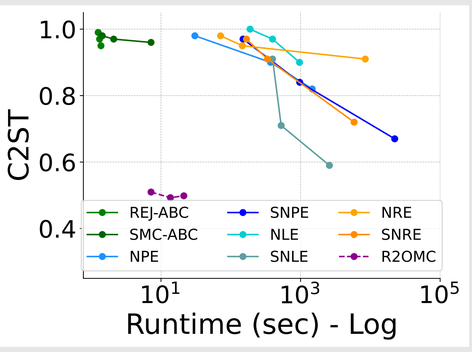
784D pixelwise-noise inference with $\sim$ 100 simulations on 10 MNIST digits.

Experiments: Lotka–Volterra & Two Moons









Lotka–Volterra (top): pairwise posterior with true $\boldsymbol{\theta}^*$ (★) within posterior mass; coverage calibration over 150 SBC repetitions follows the diagonal.
Two Moons (bottom): pairwise posterior and C2ST vs. runtime.