RandNet-Parareal: a time-parallel PDE solver using Random Neural Networks

38th Neural Information Processing Systems (NeurIPS 2024)

Guglielmo Gattiglio, Lyudmila Grigoryeva, Massimiliano Tamborrino
Department of Statistics, University of Warwick, Coventry, UK guglielmo.gattiglio@warwick.ac.uk

https://arxiv.org/abs/2411.06225

Time parallelization: a crucial technology

Parallel-in-Time methods can drive major advancements in computer-based numerical simulations in the coming years

- The scale of supercomputing keeps increasing
- Space parallelization has reached saturation, e.g. nuclear fusion [1]
- No alternative for long horizon simulation, e.g. molecular dynamics [2]

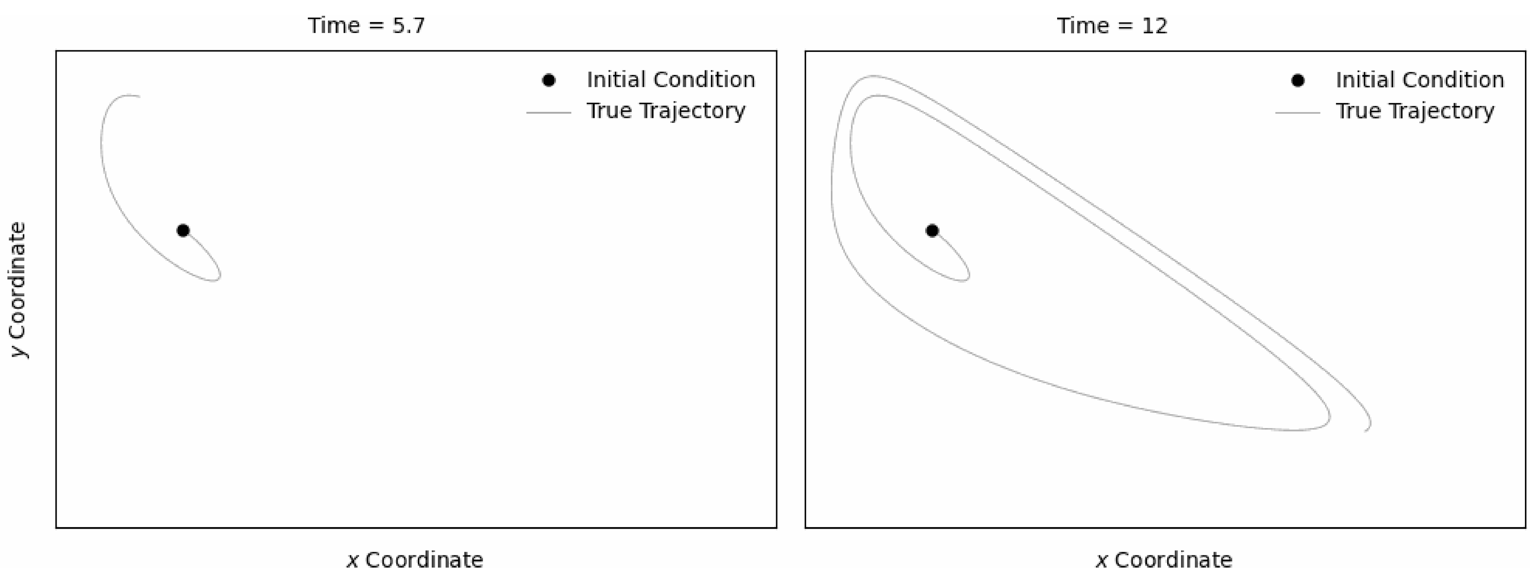
Integrating data-driven learning can drastically speed-up simulations ⟶

RandNet-Parareal at a glance

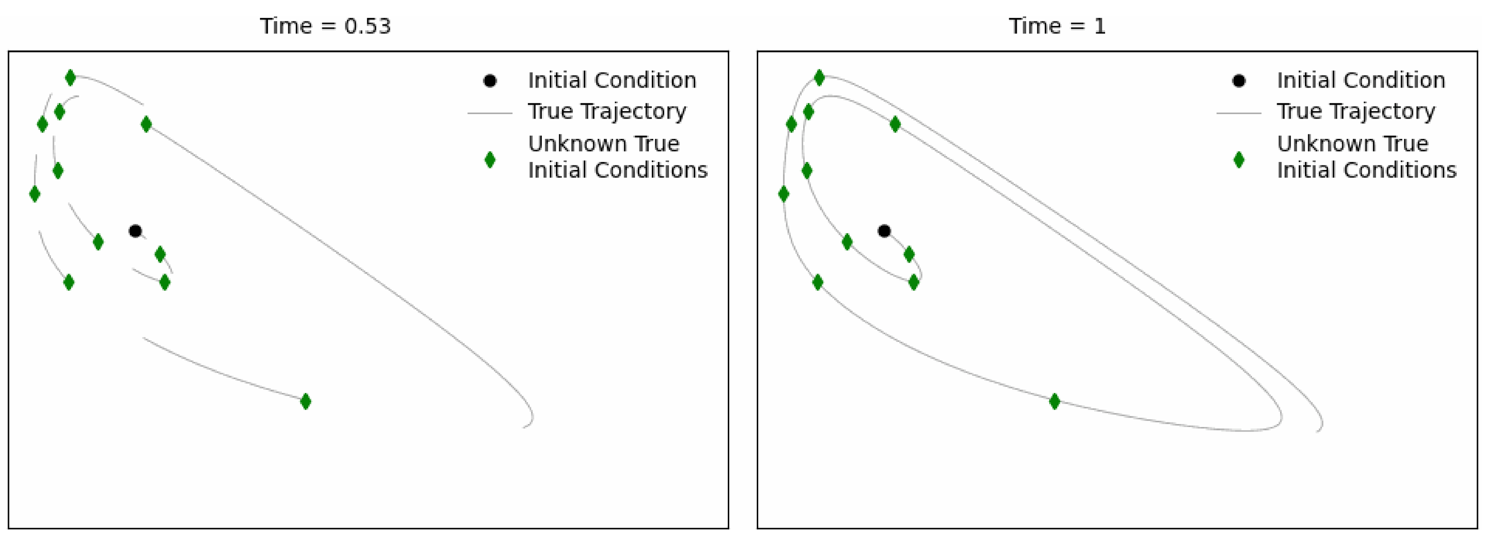
PDE System	Speed-up over Parareal	Speed-up over $\mathscr{F}$
1D Viscous Burgers'	x8.6 - x21	x12.6 - x30
2D Diffusion-Reaction	x3 - x5	x5.4 - x124
2D shallow water	x1.3 - x3.6	x16 - x39
2D & 3D Brusselator	x3.4 - x4.4	x249 - x253

Parareal & Existing Approaches

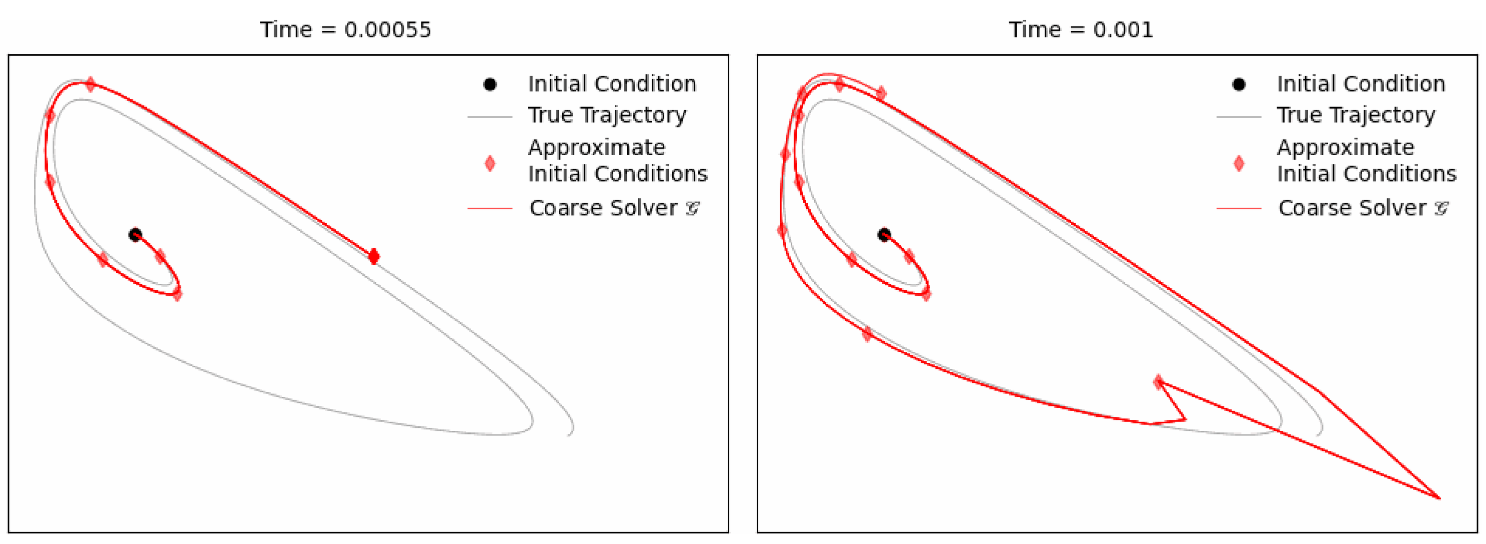
Compute the true solution *sequentially* - slow!



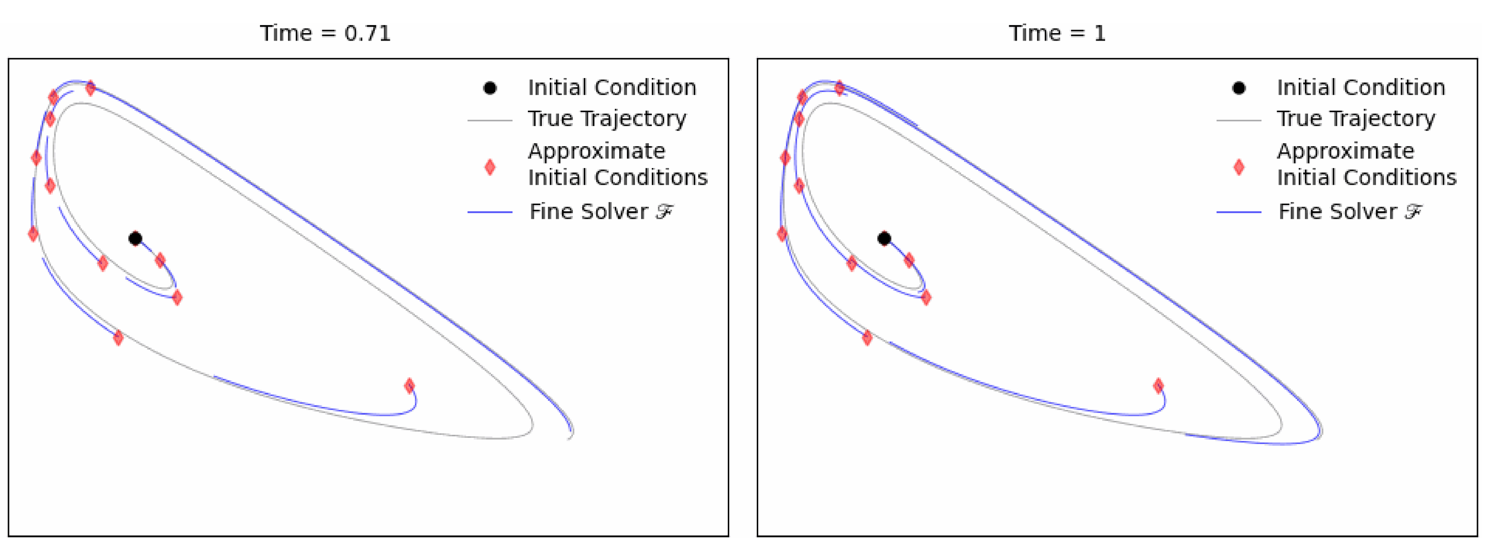
Compute the true solution on $N = 12$ intervals *in parallel* - fast!



Approximate the initial conditions *sequentially* - inaccurate but fast



Estimate the solution *in parallel* - fast but still imprecise



Let $\mathscr{F}$ be an accurate, slow numerical solver and $\mathscr{G}$ be an imprecise, fast one. Parareal [3] updates the solution U_i^k at time t_i iteration k as

$$U_i^k = \underbrace{\mathscr{G}(U_{i-1}^k)}_{\substack{\text{Sequential} \\ \mathscr{G}\text{ evolution}}} + \underbrace{(\mathscr{F}-\mathscr{G})(U_{i-1}^{k-1})}_{\mathscr{G}\text{ error correction}}$$

$(\mathscr{F}-\mathscr{G})(\cdot)$ is approximated using previous iteration data, inaccurate

$$(\mathscr{F}-\mathscr{G})(U_{i-1}^{k}) \text{ vs } (\mathscr{F}-\mathscr{G})(U_{i-1}^{k-1})$$

GParareal [4]: Approximate $\mathscr{F}-\mathscr{G}$ using Gaussian processes (GP)

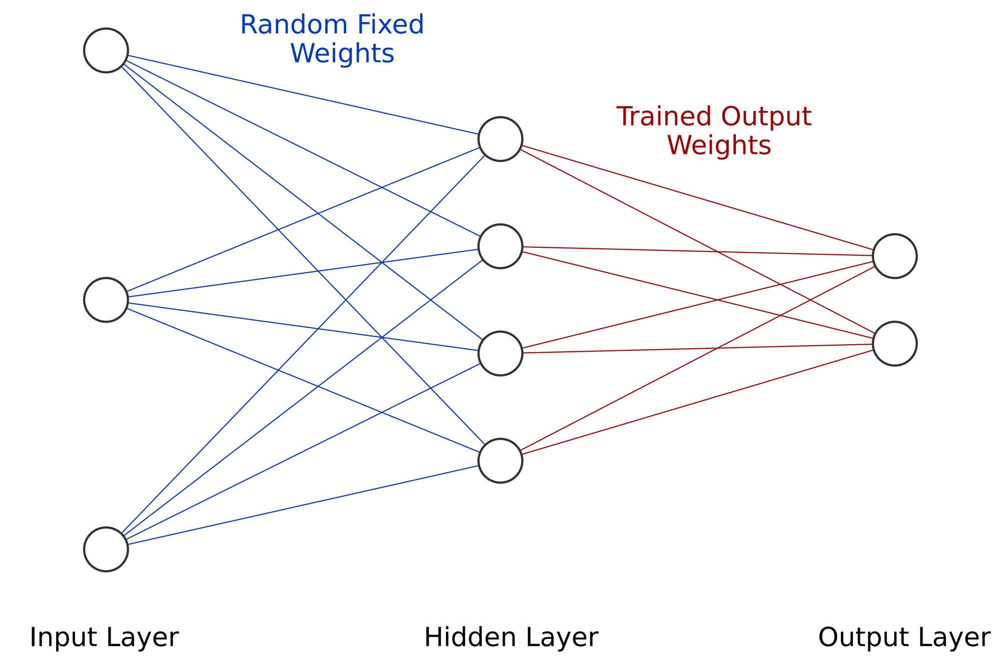
- Faster convergence but expensive to train at $O(N^3)$ cost

nnGParareal [5]: Approximate $\mathscr{F}-\mathscr{G}$ using a $\bar{k}$-nearest neighbors GP

- Reduced cost $O(\bar{k}^3), \bar{k} \ll N$, but not scalable to high-dimensions

RandNet-Parareal

A better model: *random weight neural networks* (RandNet) [6]

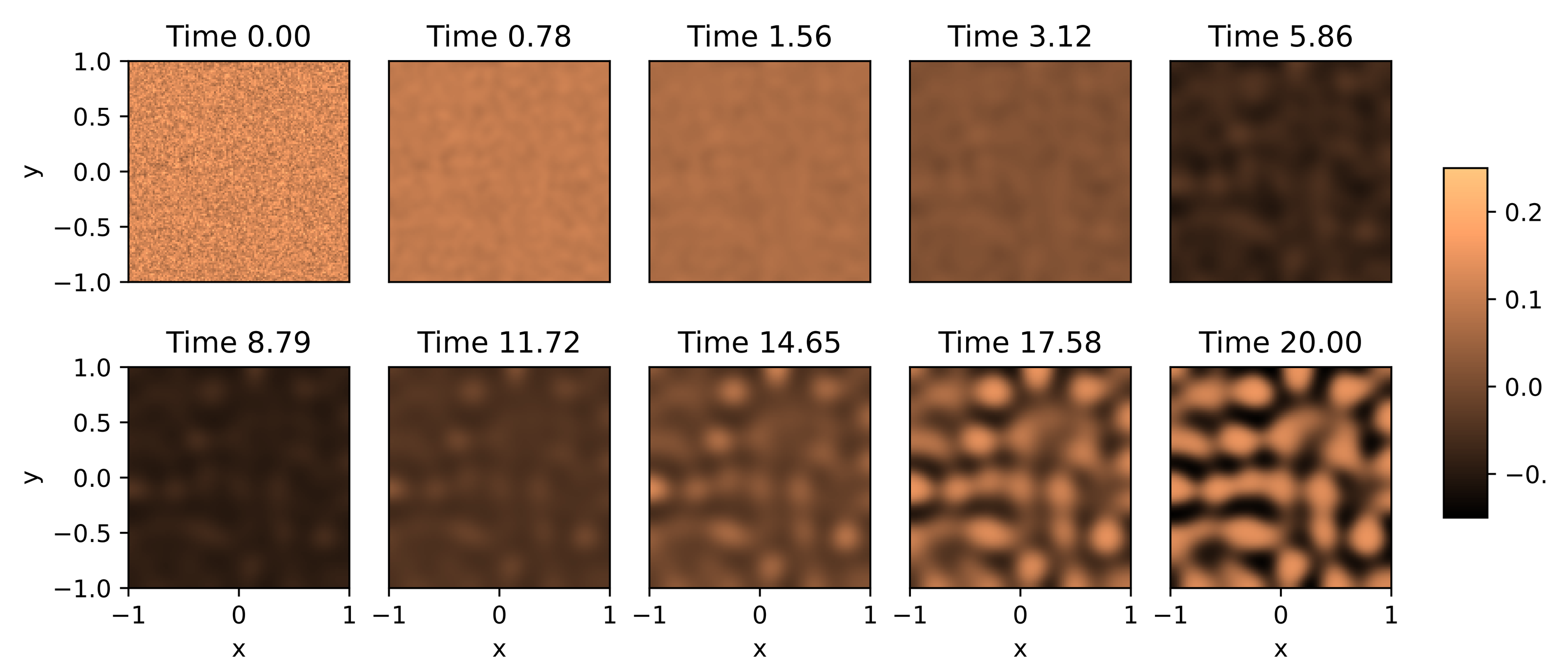


- Closed form solution for the RandNet output weights
- Universal approximator [7]
- Avoids back-propagation, stable and fast training
- Strong empirical performance

Focus: RandNet-Parareal on 2D Diffusion-Reaction

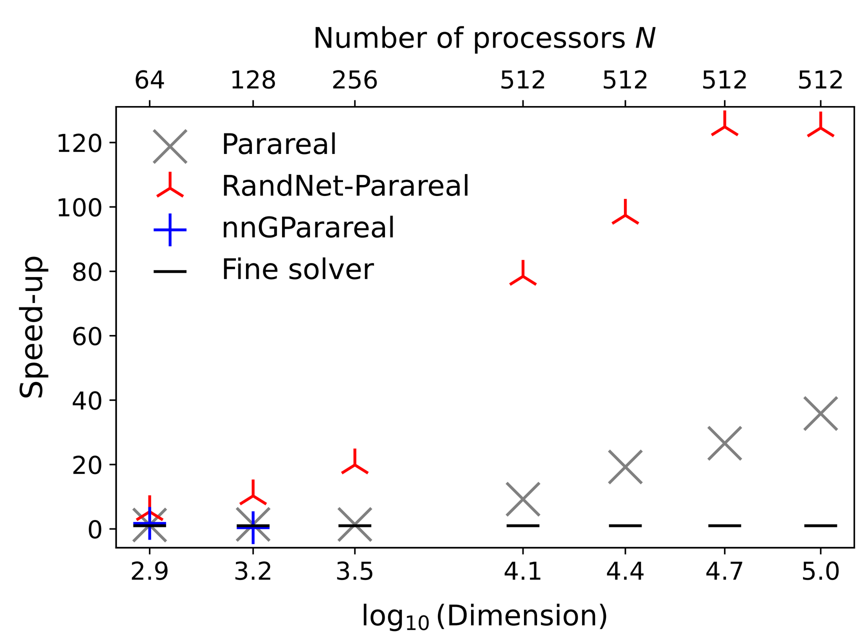
Here, $u = u(t, x, y)$ is the activator with coefficient D_u and reaction function $R_u = R_u(u, v)$. Similarly for the inhibitor $v = v(t, x, y)$

$$\partial_t u = D_u \partial_{xx} u + D_u \partial_{yy} u + R_u, \quad \partial_t v = D_v \partial_{xx} v + D_v \partial_{yy} v + R_v,$$

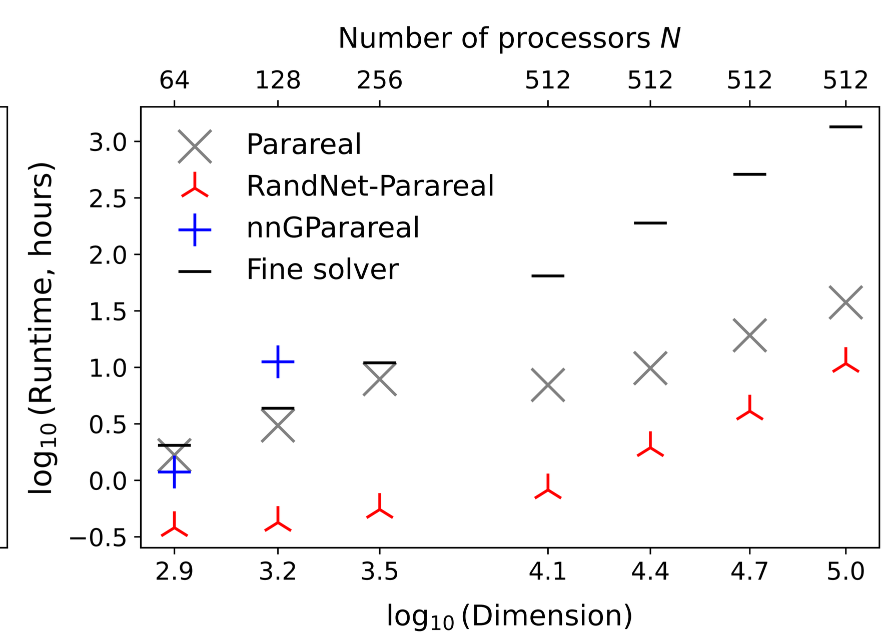


Numerical solution over $(x, y) \in [-1, 1]^2$; only u shown.

Speed-up



Runtime



References

(1) D. Samaddar, D. P. Coster et al., *Comput. Phys. Commun.*, 2019, **235**, 246–257.
(2) O. Gorynina, F. Legoll et al., *Comptes Rendus. Mecanique*, 2023, **351**, 479–503.
(3) J.-L. Lions, Y. Maday et al., *Comptes Rendus de l'Academie des Sci. I-Mathematics*, 2001, **332**, 661–668.
(4) K. Pentland, M. Tamborrino et al., *Stat. Comput.*, 2023, **33**, 23.
(5) G. Gattiglio, L. Grigoryeva et al., *arXiv Prepr.*, 2024.
(6) G.-B. Huang, Q.-Y. Zhu et al., *IEEE Int. Jt. Conf. on Neural Networks*, 2004, **2**, 985–990.
(7) L. Gonon, L. Grigoryeva et al., *The Ann. Appl. Probab.*, 2023, **33**, 28–69.