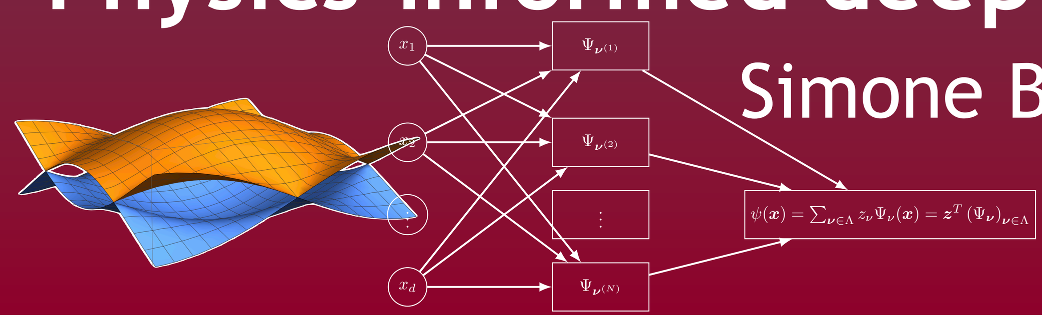
Physics-informed deep learning for high-dimensional diffusion-reaction equations



Simone Brugiapaglia[1], Nick Dexter[2], Samir Karam[1], Weiqi Wang[3]

[1]Concordia University, [2]Florida State University, [3]University of Victoria

IVADO Thematic Semester 2026, Uncertainty in AI, June 8-11

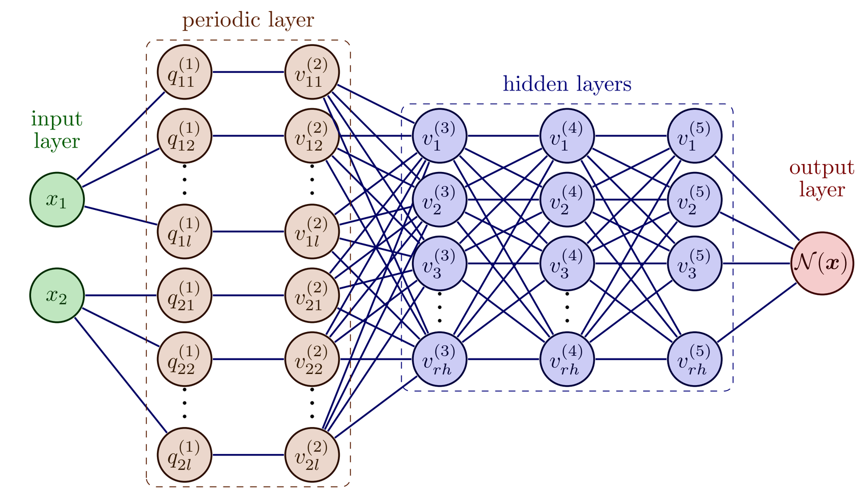
1. Deep learning for high-dimensional PDEs

In scientific machine learning, Physics-Informed Neural Networks (PINNs) have emerged as a powerful tool to solve high-dimensional Partial Differential Equations (PDEs), from computational finance (Black-Scholes) to statistical mechanics (many-electron Schrödinger equation).

- **Key challenge:** classical methods are not easily applied to high-dimensional settings requiring huge amounts of data to converge.
  $\Longrightarrow$ **exponential scaling of data.**
  ↪ **high computational cost.**
  ↪ **inefficient in modern applications.**
- **Recent results** have shown that deep neural networks (DNNs) are exceptionally good at tackling rising dimensions[7,5].

- **Goal:** approximate $u \in H^2(\mathbb{T}^d)$ s.t.
  $\mathscr{L}[u](x) := -\nabla \cdot (a(\mathbf{x})\nabla u(\mathbf{x})) + \rho u(\mathbf{x}) = f(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{T}^d$, given $a \in C^1(\mathbb{T}^d), \quad \min_{\mathbf{x}\in\mathbb{T}^d} a(\mathbf{x}) \geq a_{\min} > 0, \quad \rho > 0,$ and $f \in L^2(\mathbb{T}^d)$.

2. Setup



In this example, $h=3$, $d=2$

- **Model architecture:** with component-wise activations $\sigma^{(k)}(\mathbf{x}) = (\sigma_j^{(k)}(x_j))$,
  $\sigma_j(x_j) = x_j$ *linear* or *RePU*[6] i.e.
  $\mathrm{RePU}_\ell(x) := \max\{0, x^\ell\},\ \ell \in \mathbb{N}$.
- **Periodic layer:** enforce periodic boundary conditions.
- **Sample points:** $\mathbf{x}_1, \ldots, \mathbf{x}_m \sim \mathrm{Unif}(\mathbb{T}^d)$,
- **Train:** by minimizing regularized Root Mean Square Error (RMSE) (★).

- **In theory:**
  - **Normalize** Fourier orthonormal basis of L^2, $\quad \Psi_{\boldsymbol{\nu}} = \dfrac{\exp(2\pi i\,\boldsymbol{\nu}\cdot\mathbf{x})}{4\pi^2\|\boldsymbol{\nu}\|_2^2 + \rho/a_0}$
  - ↪ **Train** only last layer $\qquad \boldsymbol{\nu} \in \Lambda \quad (*)$
  - ↪ **Replicate** basis in rest of network.

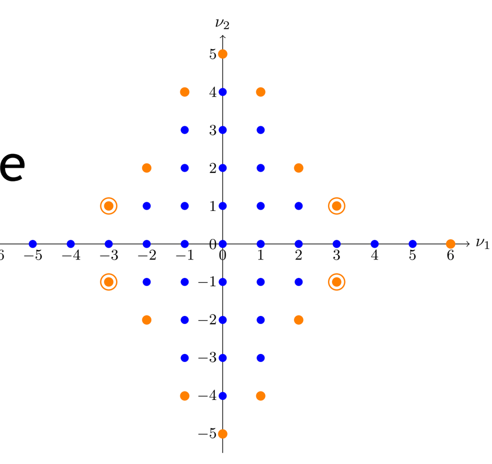
3. Compressive Fourier Collocation (CFC)[8]

Given a rescaled Fourier basis $\{\Psi_{\boldsymbol{\nu}}\}_{\boldsymbol{\nu}\in\Lambda}$, $|\Lambda| = N$, m i.i.d. points $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{T}^d$, and $m \ll N$, define

$$Az = b,$$

where $A_{ij} = \mathscr{L}[\Psi_{\boldsymbol{\nu}_j}](\mathbf{x}_i)$ and $b_i = f(\mathbf{x}_i)$. **Solve** for z using *lower Orthogonal Matching Pursuit*[7] and set

$$\hat{u} = \sum_{\boldsymbol{\nu}\in\Lambda} \hat{c}_{\boldsymbol{\nu}} \Psi_{\boldsymbol{\nu}}.$$



4. Main result (Practical Existence Theorem (PET))

Theorem. Given $d, s, n, \ell \in \mathbb{N}$, with $\ell \geq 2$, and $\epsilon \in (0,1)$, there exist:

1. a class of neural networks $\mathcal{N}$, of the form described above, s.t. $\forall \psi \in \mathcal{N}$, width($\psi$), depth($\psi$) are at most polynomial in d.
2. a regularization function $\mathscr{R} : \mathcal{N} \to [0,\infty)$,
3. a choice of tuning parameter λ depending only on a, ρ, and s,

s.t. the following holds with probability at least $1-\epsilon$. Under *suitable sufficient conditions* on the PDE parameters, let

$$m \gtrsim_{a,\rho} s\log^2(2s)\cdot\left(\min\{\log(n)+d, \log(2n)\log(2d)\} + \log(\epsilon^{-1})\right).$$

Then, every minimizer $\hat{\psi}$ of

$$(\star) \quad \min_{\psi\in\mathcal{N}} \sqrt{\frac{1}{m}\sum_{i=1}^m |\mathscr{L}[\psi](\mathbf{x}_i) - f(\mathbf{x}_i)|^2} + \lambda\mathscr{R}(\psi), \quad \text{satisfies}$$

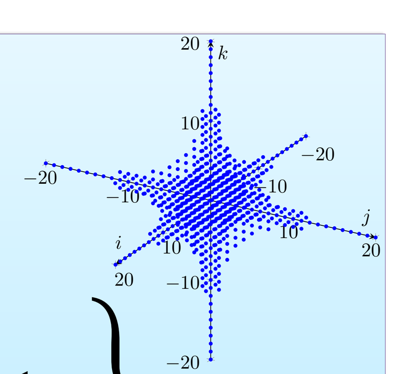
$$\|u-\hat{\psi}\|_{L^2} + \|(\Delta-\rho)(u-\hat{\psi})\|_{L^2} \lesssim_{a,\rho,d} \frac{\sigma_s(\mathbf{c}_\Lambda)_1}{\sqrt{s}} \cdot \left(\frac{\|u-u_\Lambda\|_{W^{2,\infty}}}{\sqrt{s}} + \|u-u_\Lambda\|_{H^2}\right).$$

and, if $\rho < 1$,

$$\|u-\hat{\psi}\|_{H^2} \lesssim_{a,\rho,d} \frac{\sigma_s(\mathbf{c}_\Lambda)_1}{\sqrt{s}} \cdot \left(\frac{\|u-u_\Lambda\|_{W^{2,\infty}}}{\sqrt{s}} + \|u-u_\Lambda\|_{H^2}\right).$$

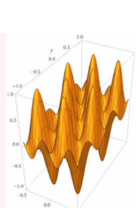
(at most linear dependence on d in the hidden constants)

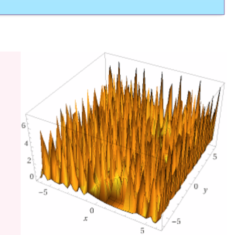
5. Proof sketch



1. Replicate $\{\Psi_{\boldsymbol{\nu}}\}_{\boldsymbol{\nu}\in\Lambda}$, while tracking width and depth of each layer. The index set is the *hyperbolic cross*[4]
$$(*) \quad \Lambda = \Lambda_{d,n}^{\mathrm{HC}} := \left\{\boldsymbol{\nu}\in\mathbb{Z}^d : \prod_{k=1}^d (|\nu_k|+1) \leq n\right\}$$
2. Show $\{\mathscr{L}[\Psi_{\boldsymbol{\nu}}]\}_{\boldsymbol{\nu}\in\Lambda}$ is *Bounded Riesz System (BRS)*[2] under sufficient conditions on a, ρ, and f.
3. BRS + lower bound on sample complexity $m \Longrightarrow$ SR-LASSO recovery guarantees $\Longrightarrow$ stable, accurate PDE solution.

6. Numerics


Ex 1: WolframAlpha


Ex 2: WolframAlpha

- Compare the performance of PINNs with CFC.

Fix a sparse diffusion function

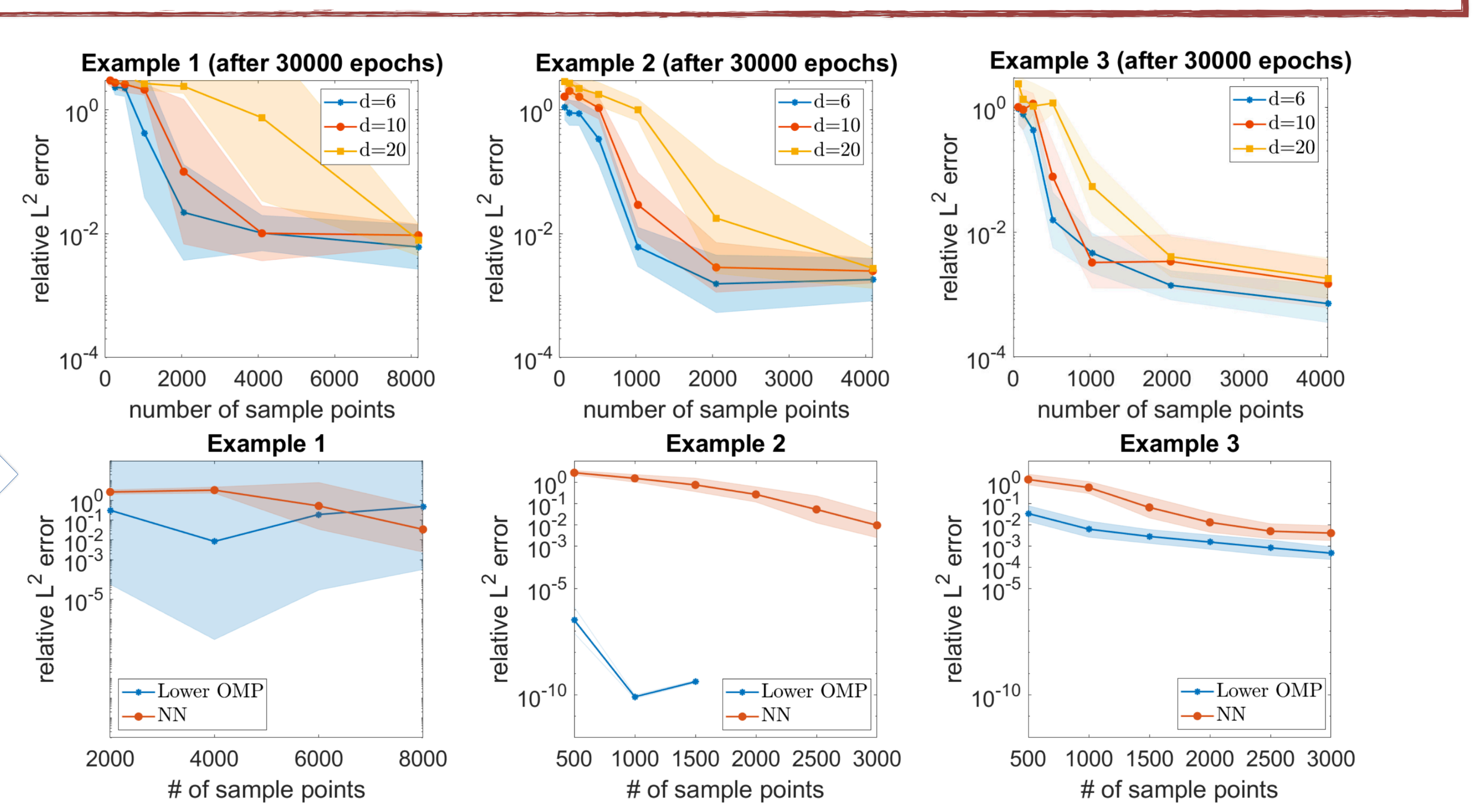
$$a(\mathbf{x}) = 1 + 0.25\sin(2\pi x_1)\sin(2\pi x_2),$$

and consider solutions

$$u_1(x) = \sin(4\pi x_1)\sin(2\pi x_2), \qquad \text{(Example 1)}$$

$$u_2(x) = \exp\left(\sin(2\pi x_1) + \sin(2\pi x_2)\right), \qquad \text{(Example 2)}$$

$$u_3(x) = \exp\left(\sum_{k=1}^d \frac{1}{k^2}\sin(2\pi x_k)\right), \qquad \text{(Example 3)}$$



7. Key takeaways

- A novel convergence result (PET) for PINNs with logarithmic, or at worst, linear scaling of the sample complexity m with dimension.
- DNNs can compete with a state-of-the-art PDE solver (CFC).
- An improvement on *uniform approximation theorems*.
  ↪These imply the existence of networks with favourable approximation properties (no sample complexity, training optimization, …)

8. Limitations

Gap between theory and practice:

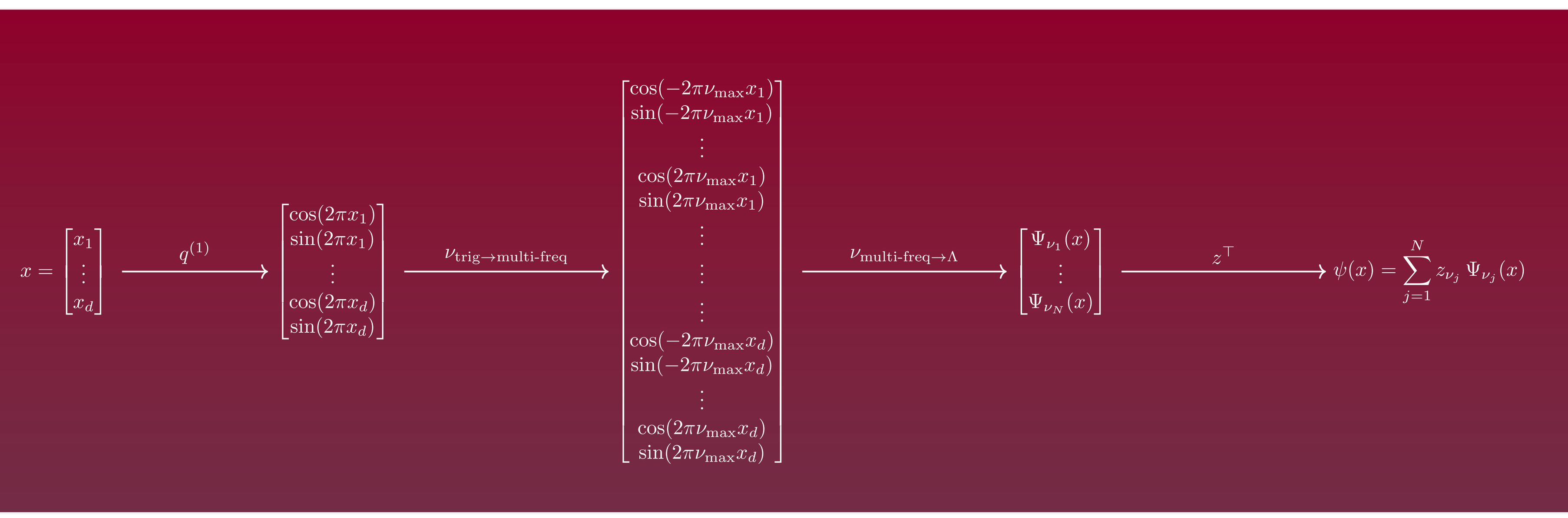
↪ **in practice**, variations of Stochastic Gradient Descent (SGD) are used to train the *whole* network; **in theory**, partial training only.

↪ **in practice**, loss is roughly minimized by SGD, but min is exact **in theory**.

$\Longrightarrow$ To replicate $\{\Psi_{\boldsymbol{\nu}}\}_{\boldsymbol{\nu}\in\Lambda}$, we use RePU and linear activations, which can exactly emulate *products*[5], but are not as standard as ReLU or tanh.

$$F_{\boldsymbol{\nu}}(\mathbf{x}) = \exp(2\pi i\boldsymbol{\nu}\cdot\mathbf{x}) = \prod_{j\in\mathrm{supp}(\boldsymbol{\nu})} \exp(2\pi i\nu_j x_j)$$

- Some open problems and improvements:
  - How to quantify the error induced by approximate min methods?
  - How to deal with switching to standard activations? (CFC, products, …)
  - How to find *necessary* sufficient conditions on the PDE parameters?



References

[1] Brugiapaglia, Dexter, SK, Wang, JMLR, 2025
[2] Brugiapaglia, Dirksen, Jung, Rauhut, ACHA, 2021
[3] Cohen, Migliorati, CCM, 2018
[4] Dũng, Temlyakov, Ullrich, Springer, 2018
[5] Han, Jentzen, E, PNAS, 2018
[6] Opschoor, Schwab, Zech, CA, 2022
[7] Raissi, Perdikaris, and Karniadakis, JCP, 2018
[8] Wang, Brugiapaglia, IMAJNA, 2024