Resolving Ill-Conditioning in Scientific Machine Learning

Marius Zeinhofer

Workshop on Structured Learning

University of Freiburg

April 30, 2026

Table of Contents

What is Scientific Machine Learning?

Scientific Machine Learning (SciML)

Simulating & inferring physical systems from **physics & data** with ML.

- **Fast Surrogate Models:** For weather, multi-scale problems, engineering[1], ...
- **Neural PDE Solvers:** For high-dimensional PDEs, like quantum[2], parametric-problems, ...
- **Discovery:** Inferring governing equations from data.[3]

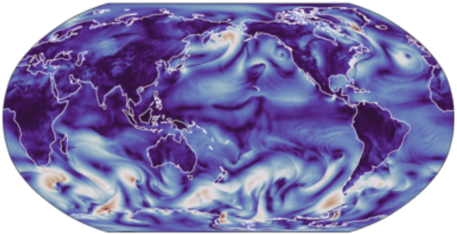


Figure: A fast neural surrogate for global weather.[4]

[1] Zongyi Li et al. (2021). "Fourier Neural Operator for Parametric Partial Differential Equations". In: *ICLR*, [2] Giuseppe Carleo and Matthias Troyer (2017). "Solving the quantum many-body problem with artificial neural networks". In: *Science*, [3] Ricky TQ Chen et al. (2018). "Neural ordinary differential equations". In: *NeurIPS* 31, [4] Boris Bonev et al. (2023). "Spherical fourier neural operators: Learning stable dynamics on the sphere". In: *ICML*.

The Language of Science: ODEs & PDEs

- Fluid mechanics

$$d_t u - \nu \Delta u + (u \cdot \nabla) u + \nabla p = f.$$

- Quantum mechanics:

$$i\hbar d_t |\psi\rangle = \hat{H} |\psi\rangle.$$

- ODEs:

$$d_t x(t) = F(t, x(t)).$$

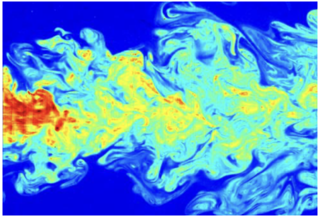


Figure: A submerged turbulent jet.[5]

SciML Examples

- **Surrogates:** Learn "solution operator": data $\mapsto$ solution.
- **PDE solution:** Approximate $u_\theta \approx u$ or $\psi_\theta \approx \psi$.
- **Discovery:** Learn $F_\theta \approx F$ from data.

[5] Source: C. Fukushima and J. Westerweel, TU Delft (CC BY 3.0)

How is SciML Different?

Priors: In the form of ODE or PDEs. Popular square loss[5] with PDE operator $\mathcal{D}$:

$$L(\theta) = \int_\Omega \underbrace{(\mathcal{D}u_\theta - f)^2}_{\text{PDE Term}} \mathrm{d}x + \underbrace{L_{\text{data}}(\theta)}_{\text{Data Fit}}$$

- **Surrogates:** Enforces constraints like conservation laws.
- **PDE Solvers:** The physics is the objective.
- **Discovery:** Embeds known laws, learning only unknowns.

Crucial Challenge: Ill-Conditioning

Physics-informed losses notoriously difficult to train.[6]

[5] Maziar Raissi et al. (2019). "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations". In: *JCP*, [6] Aditi Krishnapriyan et al. (2021). "Characterizing Possible Failure Modes in Physics-Informed Neural Networks". In: *NeurIPS*.

The Root Cause: Spectral Ill-Conditioning

- PDE operators have unbounded spectra

$$\sigma(\mathcal{D}) = \{\lambda_1 \leq \lambda_2 \leq \cdots \to \infty\}.$$

- Leads to stiff physics-informed losses.
- Optimization notoriously difficult. Even practically impossible?[7]

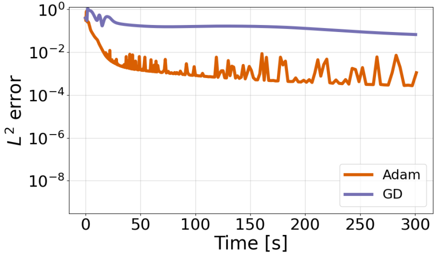


Figure: Slow convergence of GD/Adam for Poisson equation.

Central Question

Is accurate physics-informed learning at all possible?

[7] Philipp Grohs and Felix Voigtlaender (2024). "Proof of the theory-to-practice gap in deep learning via sampling complexity bounds for neural network approximation spaces". In: *Foundations of Computational Mathematics* 24.4, pp. 1085–1143.

Preconditioned Gradient Descent

- Classically: Handle ill-conditioning via preconditioning.

$$\theta_{k+1} = \theta_k - \eta_k P(\theta_k)^{-1} \nabla L(\theta_k)$$

- Correct spectrum of PDE operator $\mathcal{D}$:

$$P(\theta)_{ij} = \int_\Omega \mathcal{D}[\partial_i u_\theta] \mathcal{D}[\partial_j u_\theta] \mathrm{d}x.$$

- We will show optimality of $P(\theta)$.

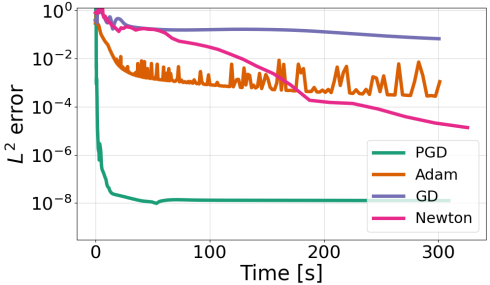


Figure: Success of preconditioned gradient descent (PGD) for the solution of a Poisson equation.

Key contribution: **Machine precision PINNs.** See '23 ICML paper[8]

[8] Johannes Müller and **Z.** (2023). "Achieving High Accuracy with PINNs via Energy Natural Gradients". In: *ICML*.

Table of Contents

Geometric Framework: Optimize-and-Project

Optimize-and-Project

(i) **Continuous Formulation:** Formulate problem in a Hilbert space $\mathcal{H}$

$$E : \mathcal{H} \to \mathbb{R}.$$

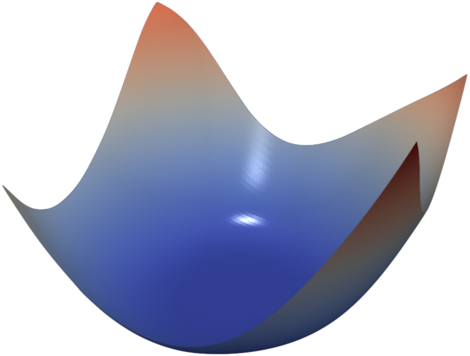
(ii) **Optimize:** Decide for an appropriate iterative algorithm *in function space* $\mathcal{H}$

$$u_{k+1} = u_k + d_k.$$

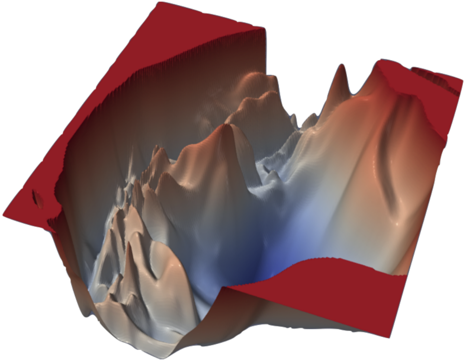
(iii) **Project:** Project d_k on the tangent space of neural network ansatz. Yields

$$\theta_{k+1} = \theta_k + \delta\theta_k \quad \text{with} \quad \nabla_\theta u_{\theta_k} \cdot \delta\theta_k \approx d_k$$

Optimize-and-Project for a Quadratic Objective I



$E(u) = \frac{1}{2}\|\Delta u + f\|^2_{L^2(\Omega)}$
Quadratic in function space



$L(\theta) = \frac{1}{2}\|\Delta u_\theta + f\|^2_{L^2(\Omega)}$
Nonconvex in parameter space

Picture adapted from Hao Li et al. (2018). "Visualizing the loss landscape of neural nets". In: *NeurIPS*.

Optimize-and-Project for a Quadratic Objective II

- **Continuous Formulation:** We use the exemplary energy (Poisson equation)

$$\min_{u \in \mathcal{H}} E(u) = \frac{1}{2} \|\Delta u + f\|_{L^2(\Omega)}^2 = \frac{1}{2} \int_\Omega (\Delta u + f)^2 \, \mathrm{d}x.$$

- **Optimize using Newton:** Optimal choice is ∞-dimensional Newton:

$$u^* = u_0 \underbrace{- E''(u_0)^{-1}[E'(u_0)]}_{=d}$$

The direction d satisfies the equation

$$\int_\Omega \Delta(u_0 + d) \Delta\varphi \, \mathrm{d}x = -\int_\Omega f \Delta\varphi \, \mathrm{d}x, \quad \text{for all } \varphi \in \mathcal{H}.$$

Optimize-and-Project for a Quadratic Objective III

- **Project:** The tangent space of the ansatz is given by

$$\operatorname{span}\{\partial_1 u_\theta, \dots, \partial_p u_\theta\}.$$

Use it as a Galerkin space to approximate the direction d:

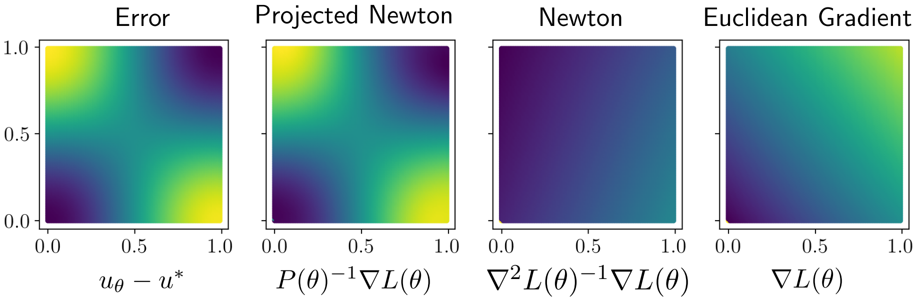
$$E'(u_\theta) \Rightarrow \nabla L(\theta), \qquad [\nabla L(\theta)]_i = \int_\Omega \Delta(u_\theta + f)\partial_i u_\theta \,\mathrm{d}x$$

$$E''(u_\theta) \Rightarrow P(\theta), \qquad [P(\theta)]_{ij} = \int_\Omega \Delta\partial_i u_\theta \Delta\partial_j u_\theta \,\mathrm{d}x$$

- **Translation:** ∞-dimensional algorithm $\Rightarrow$ tractable **projected Newton** scheme.

$$u_{k+1} = u_k - E''(u_k)^{-1}[E'(u_k)] \quad \Rightarrow \quad \theta_{k+1} = \theta_k - \eta_k P(\theta)^\dagger \nabla L(\theta_k)$$

Optimality of the Preconditioner



Projection Property: Quadratic Energy E

Here, Proj_k is projection on tangent space and $\varepsilon_k = \mathcal{O}(\eta_k^2 \|P(\theta_k)^{-1}\nabla L(\theta_k)\|^2)$

$$u^* = u_0 - \underbrace{E''(u_0)^{-1}[E'(u_0)]}_{=u_0 - u^* \text{(optimal direction)}} \quad \Rightarrow \quad u_{\theta_{k+1}} = u_{\theta_k} - \eta_k \underbrace{\mathrm{Proj}_k[u_{\theta_k} - u^*]}_{\text{proj. optimal direction}} + \varepsilon_k.$$

Result: Efficient Projected Newton Method

- Computational scheme

$$\theta_{k+1} = \theta_k - \eta_k P(\theta)^\dagger \nabla L(\theta_k),$$

with $P(\theta) = \int_\Omega \Delta\partial_i u_\theta \Delta\partial_j u_\theta \,\mathrm{d}x$.

- Interpretations:
    - Projected Newton
    - Natural Gradient
    - Generalized Gauss-Newton

$$\nabla^2 L(\theta)_{ij} = P(\theta)_{ij} + E'(u_\theta)[\partial_i\partial_j u_\theta].$$

- Details in our 23' ICML paper[8]

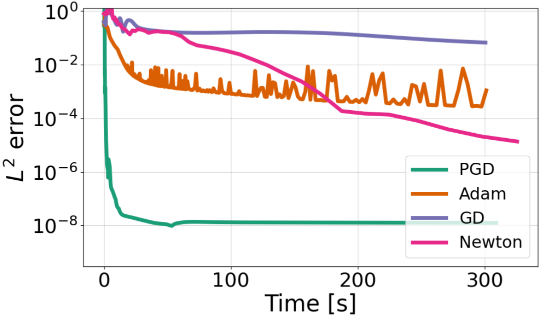


Figure: Success of the projected Newton method (PGD) for the solution of a Poisson equation.

[8] Johannes Müller and **Z.** (2023). "Achieving High Accuracy with PINNs via Energy Natural Gradients". In: *ICML*.

Broader Perspective

Meta-principle for Algorithm Design

High accuracy for physics-informed learning, see '24 ICML position paper[9]

- **General Objectives:** Not restricted to quadratic E.
- **General Optimizer:** Not restricted to Newton's method in function space.
- **Further Work:**
  - Design & benchmarking of optimizers[10,11,12].
  - A priori & a posteriori analysis for neural PDE solvers[13,14,15] .

[9] Johannes Müller and **Z.** (2024). "Position: Optimization in SciML Should Employ the Function Space Geometry". In: *ICML*, [10] Anas Jnini, Flavio Vella, and **Z.** (2025). "Gauss-Newton Natural Gradient Descent for Physics-Informed Computational Fluid Dynamics". In: *Computers & Fluids*, [11] **Z.**, Rami Masri, and Kent-Andre Mardal (2024). "A Unified Framework for the Error Analysis of Physics-Informed Neural Networks". In: *IMA Journal on Numerical Analysis*, [12] Felix Dangel, Johannes Müller, and **Z.** (2024). "Kronecker-Factored Approximate Curvature for Physics-Informed Neural Networks". In: *NeurIPS*, [13] **Z.**, Rami Masri, and Kent-Andre Mardal (2024). "A Unified Framework for the Error Analysis of Physics-Informed Neural Networks". In: *IMA Journal on Numerical Analysis*, [14] Patrick Dondl, Johannes Müller, and **Z.** (2022). "Uniform Convergence Guarantees for the Deep Ritz Method for Nonlinear Problems". In: *Advances in Continuous and Discrete Models*, [15] Alex Kaltenbach and **Z.** (2025). "The Deep Ritz Method for Parametric p-Dirichlet Problems". In: *Advances in Continuous and Discrete Models*.

Table of Contents

The Variational Monte Carlo Method

- **Goal:** Find ground state of quantum system.
- **Strategy:** For Hamiltonian $\hat{H}$, optimize neural wavefunction ψ_θ via loss[16,17,18]

$$E(\psi_\theta) = \frac{\langle \psi_\theta \mid \hat{H} \mid \psi_\theta \rangle}{\langle \psi_\theta \mid \psi_\theta \rangle} = \mathbb{E}_{|\psi|^2}\left[\frac{\hat{H}\psi_\theta}{\psi_\theta}\right]$$

- **Curse of Dimensionality:** $\psi : \mathbb{R}^{3N} \to \mathbb{R}$:
  - 10 electrons on a 100-point grid: space for ψ has 10^{30} dimensions.
  - Not even matvecs with $\hat{H}$ possible!

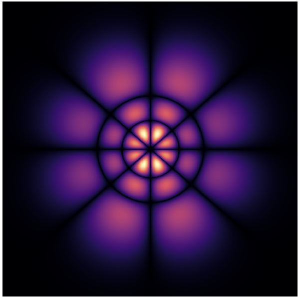


Figure: Excited state H-orbital. Picture from https://github.com/liam-ilan/electron-orbitals.

[16] Giuseppe Carleo and Matthias Troyer (2017). "Solving the quantum many-body problem with artificial neural networks". In: *Science*, [17] David Pfau et al. (2020). "Ab Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks". In: *Physical Review Research*, [18] Michael Scherbela et al. (2025). "Accurate ab-initio neural-network solutions to large-scale electronic structure problems". In: *arXiv preprint arXiv:2504.06087*.

Optimize-and-Project: Riemannian GD I

- **Continuous Formulation:** Minimize E over sphere $\mathbb{S} = \{\psi \mid \|\psi\|^2 = 1\}$

$$\min_{\psi \in \mathbb{S}} E(\psi) = \langle \psi \mid \hat{H} \mid \psi \rangle$$

- **Optimize:** Riemannian gradient descent

$$\psi_{k+1} = \frac{\psi_k - \eta_k \operatorname{grad}_{\mathbb{S}} E(\psi_k)}{\|\psi_k - \eta_k \operatorname{grad}_{\mathbb{S}} E(\psi_k)\|},$$

where the gradient on $\mathbb{S}$ is given by

$$\operatorname{grad}_{\mathbb{S}} E(\psi) = 2[\hat{H}\psi_k - E(\psi)\psi] \in T_\psi \mathbb{S}.$$

Optimize-and-Project: Riemannian GD II

- **Project:** Galerkin project $\operatorname{grad}_{\mathbb{S}} E(\psi_k)$ into the space spanned by

$$\partial_i \hat{\psi}_\theta := \partial_i \left(\frac{\psi_\theta}{\|\psi_\theta\|} \right) = \frac{1}{\|\psi_\theta\|} \left(I - \hat{\psi}_\theta \otimes \hat{\psi}_\theta \right) \partial_i \psi_\theta \in T_{\psi_\theta} \mathbb{S}$$

Result: Stochastic Reconfiguration

Projected Riemannian Gradient Descent is "stochastic reconfiguration"[19]

$$\theta_{k+1} = \theta_k - \eta_k S(\theta_k)^{-1} \nabla_\theta E(\theta_k), \quad k = 0, 1, 2 \ldots$$

where

$$S(\theta_k)_{ij} = \langle \partial_i \hat{\psi}_\theta \mid \partial_j \hat{\psi}_\theta, \rangle_{L^2}, \quad \partial_i E(\theta_k) = 2 \langle \partial_i \hat{\psi}_\theta \mid \hat{H} \mid \hat{\psi}_\theta \rangle.$$

[19] Sandro Sorella (1998). "Green function Monte Carlo with stochastic reconfiguration". In: *Physical review letters* 80.20, p. 4558.

Stochastic Reconfiguration (SR)

- **Insight:** Projected RGD is SR.
- **SOTA Method:** Used in most recent works.
- **Slow:** Linear Convergence

$$\|\psi_k - \psi^*\| = \mathcal{O}\left(\left|1 - \frac{1}{2}\frac{E_1 - E_0}{E_{\max} - E_0}\right|^k\right)$$

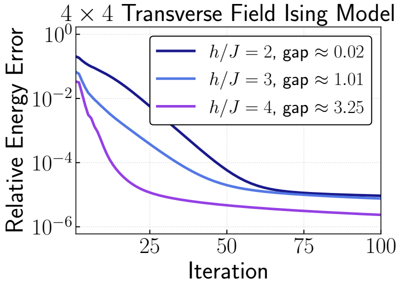


Figure: SR convergence for varying spectral gaps.

Can we design projected eigenvalue methods?
Optimize-and-project provides the framework!

Optimize-and-Project: Riemannian Newton I

- **Continuous Formulation:** Minimize E over sphere $\mathbb{S} = \{\psi \mid \|\psi\|^2 = 1\}$

$$\min_{\psi \in \mathbb{S}} E(\psi) = \langle \psi \mid \hat{H} \mid \psi \rangle$$

- **Optimize:** Globalized Riemannian Newton with modified Hessian:

$$\mathcal{H}_\tau(v, w) = \langle v \mid \hat{H} \mid w \rangle + \tau \langle v \mid w \rangle, \quad v, w \in T_\psi \mathbb{S}.$$

The scheme is

$$\psi_{k+1} = \frac{\psi_k - \mathcal{H}_\tau^{-1}[\operatorname{grad}_{\mathbb{S}} E(\psi_k)]}{\|\psi_k - \mathcal{H}_\tau^{-1}[\operatorname{grad}_{\mathbb{S}} E(\psi_k)]\|}.$$

Note:

- **Riemannian Newton:** If $\tau = -E(\psi)$, it holds $\mathcal{H}_\tau = \operatorname{Hess}_{\mathbb{S}} E(\psi)$.
- **Globalization:** If $\tau > |E_0|$ the form $\mathcal{H}_\tau$ is positive definite.

Interpretation as Inverse Iteration

Lemma (Globalized Riemannian Newton is Inverse Iteration)

Assume that $\tau \notin \sigma(\hat{H})$. Then it holds

$$\frac{\psi_k - \mathcal{H}_\tau^{-1}[\operatorname{grad}_{\mathbb{S}} E(\psi_k)]}{\|\psi_k - \mathcal{H}_\tau^{-1}[\operatorname{grad}_{\mathbb{S}} E(\psi_k)]\|} = \frac{(\hat{H} - \tau\,\mathrm{Id})^{-1}\psi_k}{\|(\hat{H} - \tau\,\mathrm{Id})^{-1}\psi_k\|.}$$

The right-hand side is the classical eigensolver **shifted Inverse Iteration**.

- If τ closest to E_0, convergence is $\mathcal{O}\left(\left|1 - \frac{E_1 - E_0}{E_1 - \tau}\right|^k\right)$.
- If $\tau = -E(\psi_k)$, we recover **Rayleigh Quotient Iteration**, with cubic convergence.

Details in recent preprint[20] with ETH D-PHYS.

[20] Victor Armegioiu et al. (2025). "Functional neural wavefunction optimization". In: *arXiv preprint arXiv:1234.45678*.

Optimize-and-Project: Riemannian Newton II

- **Project:** Galerkin project $\mathcal{H}_\tau$ and $\operatorname{grad}_{\mathbb{S}} E(\psi_k)$ into the space spanned by

$$\partial_i \hat{\psi}_\theta \coloneqq \partial_i \left(\frac{\psi_\theta}{\|\psi_\theta\|} \right) = \frac{1}{\|\psi_\theta\|} \left(I - \hat{\psi}_\theta \otimes \hat{\psi}_\theta \right) \partial_i \psi_\theta \in T_{\psi_\theta}\mathbb{S}$$

Result: Projected Inverse Iteration

Novel method **Projected Inverse Iteration (PII)**.

$$\theta_{k+1} = \theta_k - \eta_k (H(\theta_k) - \tau S(\theta_k))^{-1} \nabla_\theta E(\theta_k), \quad k = 0, 1, 2 \dots$$

where

$$H(\theta_k)_{ij} = \hat{H}(\partial_i \hat{\psi}_\theta, \partial_j \hat{\psi}_\theta) \quad S(\theta_k)_{ij} = \langle \partial_i \hat{\psi}_\theta \mid \partial_j \hat{\psi}_\theta, \rangle_{L^2}.$$

Projected Inverse Iteration

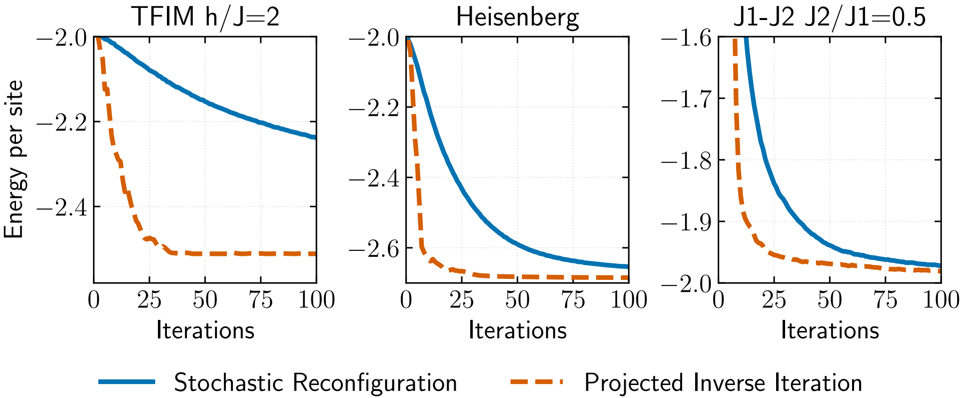
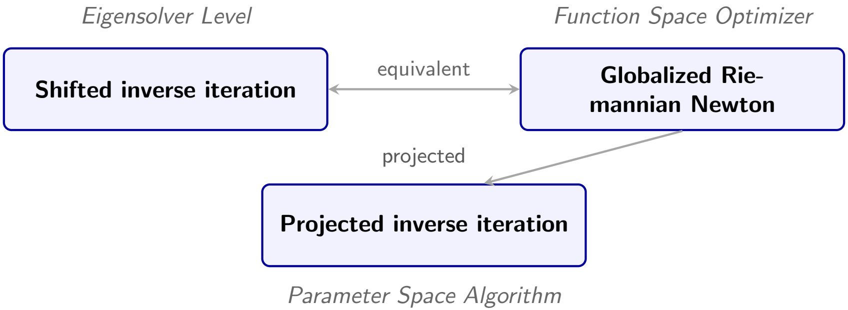


Figure: SR vs PII for three 10×10 systems.

Summary



- Neural Quantum State optimization is a fast-moving field[21,22,23]
- Our geometrical viewpoint provides anchor point.

[21] Du Jiang et al. (2025). "Neural scaling laws surpass chemical accuracy for the many-electron Schrödinger equation". In: *arXiv:2508.02570*, [22] Michael Scherbela et al. (2025). "Accurate ab-initio neural-network solutions to large-scale electronic structure problems". In: *arXiv preprint arXiv:2504.06087*, [23] Yuntian Gu et al. (2025). "Solving the hubbard model with neural quantum states". In: *arXiv preprint arXiv:2507.02644*.

Table of Contents

Computational Challenges

Challenges

What are the computational bottlenecks?

- **Computation of high-order input derivatives** to evaluate loss functions:

$$L(\theta) = \frac{1}{2}\|\Delta u_\theta - f\|^2_{L^2(\Omega)}$$

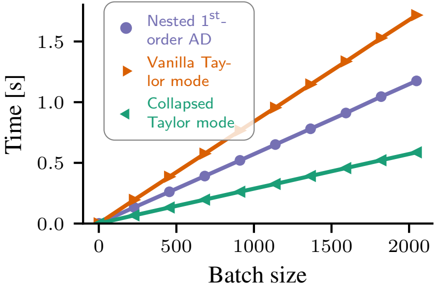
- **Dense linear solves** to compute Galerkin projections for optimization:

$$\theta_{k+1} = \theta_k - \eta_k G(\theta_k)^\dagger \nabla L(\theta_k), \quad k = 0, 1, 2 \ldots$$

$$G(\theta)_{ij} = \int_\Omega \Delta\partial_{\theta_i} u_\theta \Delta\partial_{\theta_j} u_\theta \, \mathrm{d}x$$

Collapsing Taylor-Mode AD[1]

- Efficient loss evaluation important.
- Bottleneck AD: Nested backprop inefficient.
- We introduce *Collapsed Taylor Mode* to accelerate PDE operator computation and reduce memory.
- Accelerates Laplacians, Bi-Laplacians, their stochastic estimators, and more.



Use Collapsed Taylor Mode

Faster, more memory efficient, and applicable to many PDE operators!

[1] Felix Dangel, Tim Siebert, et al. (2025). "Collapsing Taylor Mode Automatic Differentiation". In: *NeurIPS*.

Scaling Dense Linear Solves

Example: Projected Newton's Method

In every step the following matrix needs to be assembled and inverted

$$G(\theta)_{ij} = D^2E(u_\theta)(\partial_{\theta_i}u_\theta, \partial_{\theta_j}u_\theta), \quad i, j = 1, \ldots, P.$$

Typically, the matrix is **dense, rank deficient and ill-conditioned**, complexity $\mathcal{O}(P^3)$.

- Option A: KFAC[2]—Approximate the matrix G such that it is easily invertible.
- Option B: Woodbury[3]—Exploit low-rank structure in overparametrized regime.

[2] Felix Dangel, Johannes Müller, and **Z.** (2024). "Kronecker-Factored Approximate Curvature for Physics-Informed Neural Networks". In: *NeurIPS*.

[3] Andrés Guzmán-Cordero et al. (2025). "Improving Energy Natural Gradient Descent through Woodbury, Momentum, and Randomization". In: *NeurIPS*.

Option A: Kronecker-factored Approximate Curvature[4]

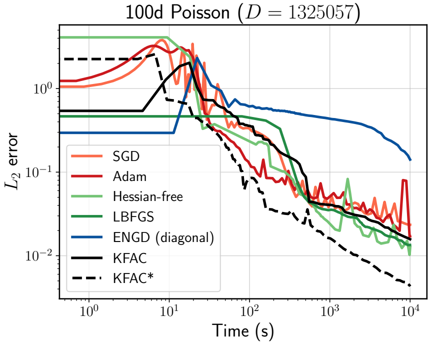
- KFAC approximation

$$
\begin{aligned}
G(\theta)^{-1} &\approx \operatorname{diag}(G_{(1)}, \dots, G_{(L)})^{-1} \\
&\approx \operatorname{diag}(A_{(1)} \otimes B_{(1)}, \dots, A_{(L)} \otimes B_{(L)})^{-1} \\
&= \operatorname{diag}(A_{(1)}^{-1} \otimes B_{(1)}^{-1}, \dots, A_{(L)}^{-1} \otimes B_{(L)}^{-1}).
\end{aligned}
$$

- Linearity yields Kronecker-structure

$$
J_W(W \mapsto Wx) = x^\top \otimes \mathrm{Id}
$$

- Treat input derivatives via Taylor mode.



KFAC

Scalable & efficient but architecture dependent.

[4] Felix Dangel, Johannes Müller, and **Z.** (2024). "Kronecker-Factored Approximate Curvature for Physics-Informed Neural Networks". In: *NeurIPS*.

Option B: Exploit Low-Rank Structure via Woodbury Identity[6]

- Common structure of natural gradient systems, with $O \in \mathbb{R}^{N \times P}$

$$d = \underbrace{(O^\top O + \varepsilon\, \mathrm{Id})^{-1}}_{\in \mathbb{R}^{P \times P}} O^\top r, \qquad \mathcal{O}(P^3)$$

- For overparametrized systems $N < P$

$$d = O^\top \underbrace{(O O^\top + \varepsilon\, \mathrm{Id})^{-1}}_{\in \mathbb{R}^{N \times N}} r, \qquad \mathcal{O}(N^2 P)$$

- Combine with: *Randomization on the kernel*, momentum-type methods[5].

Simple and effective

Rivals performance of K-FAC, but architecture agnostic.

[5] Gil Goldshlager, Nilin Abrahamsen, and Lin Lin (2024). "A Kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions". In: *JCP*.

[6] Andrés Guzmán-Cordero et al. (2025). "Improving Energy Natural Gradient Descent through Woodbury, Momentum, and Randomization". In: *NeurIPS*.

Conclusion

Function Space Viewpoint

Blueprint for the design of optimization methods for loss functions involving PDE terms and nonlinear ansatz spaces[7].

- Geometric viewpoint is promising and generally applicable.
- Many open questions: Momentum, PDE constrained optimization, ...
- **Thank you for your attention!**

[7] Johannes Müller and **Z.** (2024). "Position: Optimization in SciML Should Employ the Function Space Geometry". In: *ICML*.

References I

- al., Maziar Raissi et (2019). “Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations”. In: *JCP*.
- Armegioiu, Victor et al. (2025). “Functional neural wavefunction optimization”. In: *arXiv preprint arXiv:1234.45678*.
- Bonev, Boris et al. (2023). “Spherical fourier neural operators: Learning stable dynamics on the sphere”. In: *ICML*.
- Carleo, Giuseppe and Matthias Troyer (2017). “Solving the quantum many-body problem with artificial neural networks”. In: *Science*.
- Chen, Ricky TQ et al. (2018). “Neural ordinary differential equations”. In: *NeurIPS* 31.
- Dangel, Felix, Johannes Müller, and **Z.** (2024). “Kronecker-Factored Approximate Curvature for Physics-Informed Neural Networks”. In: *NeurIPS*.

References II

- Dangel, Felix, Tim Siebert, et al. (2025). “Collapsing Taylor Mode Automatic Differentiation”. In: *NeurIPS*.
- Dondl, Patrick, Johannes Müller, and **Z.** (2022). “Uniform Convergence Guarantees for the Deep Ritz Method for Nonlinear Problems”. In: *Advances in Continuous and Discrete Models*.
- Goldshlager, Gil, Nilin Abrahamsen, and Lin Lin (2024). “A Kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions”. In: *JCP*.
- Grohs, Philipp and Felix Voigtlaender (2024). “Proof of the theory-to-practice gap in deep learning via sampling complexity bounds for neural network approximation spaces”. In: *Foundations of Computational Mathematics* 24.4, pp. 1085–1143.
- Gu, Yuntian et al. (2025). “Solving the hubbard model with neural quantum states”. In: *arXiv preprint arXiv:2507.02644*.
- Guzmán-Cordero, Andrés et al. (2025). “Improving Energy Natural Gradient Descent through Woodbury, Momentum, and Randomization”. In: *NeurIPS*.

References III

- Jiang, Du et al. (2025). "Neural scaling laws surpass chemical accuracy for the many-electron Schrödinger equation". In: *arXiv:2508.02570.*
- Jnini, Anas, Flavio Vella, and **Z.** (2025). "Gauss-Newton Natural Gradient Descent for Physics-Informed Computational Fluid Dynamics". In: *Computers & Fluids.*
- Kaltenbach, Alex and **Z.** (2025). "The Deep Ritz Method for Parametric p-Dirichlet Problems". In: *Advances in Continuous and Discrete Models.*
- Krishnapriyan, Aditi et al. (2021). "Characterizing Possible Failure Modes in Physics-Informed Neural Networks". In: *NeurIPS.*
- Li, Hao et al. (2018). "Visualizing the loss landscape of neural nets". In: *NeurIPS.*
- Li, Zongyi et al. (2021). "Fourier Neural Operator for Parametric Partial Differential Equations". In: *ICLR.*
- Müller, Johannes and **Z.** (2023). "Achieving High Accuracy with PINNs via Energy Natural Gradients". In: *ICML.*

References IV

- Müller, Johannes and **Z.** (2024). "Position: Optimization in SciML Should Employ the Function Space Geometry". In: *ICML*.
- Pfau, David et al. (2020). "Ab Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks". In: *Physical Review Research*.
- Scherbela, Michael et al. (2025). "Accurate ab-initio neural-network solutions to large-scale electronic structure problems". In: *arXiv preprint arXiv:2504.06087*.
- Sorella, Sandro (1998). "Green function Monte Carlo with stochastic reconfiguration". In: *Physical review letters* 80.20, p. 4558.
- **Z.**, Rami Masri, and Kent-Andre Mardal (2024). "A Unified Framework for the Error Analysis of Physics-Informed Neural Networks". In: *IMA Journal on Numerical Analysis*.