CHONKNORIS

Operator Learning at Machine Precision

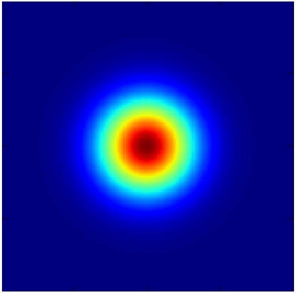
Presented by Joseph McLaughlin for UO CS Journal Club; Topics on machine learning, scientific computing. Spring 2026 Session. Friday, April 10, 2026.

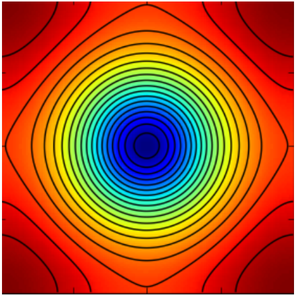
Operator Learning

- Learn a mapping $G : U \mapsto V$ between function spaces.
  - Input a function $u \in U$
    - Could be material properties or initial conditions
  - Output a function $v \in V$
    - Usually a solution to the PDE problem
- G is typically defined implicitly for $F(u, G(u)) = 0$ where F is known

Operator Learning: Intuition

- Poisson equation: $-v''(x) = u(x)$ on $[0,1]$, with $v(0) = v(1) = 0$
  - $u(x)$ is the forcing term – where and how hard you "push"
  - $v(x)$ is the response – the shape that results



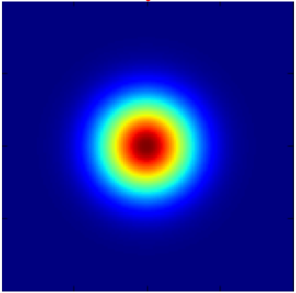


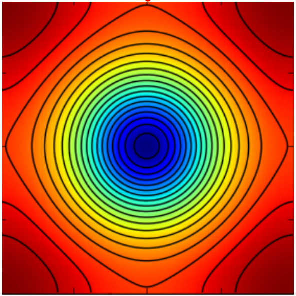
Operator Learning: Intuition

- Poisson equation: $-v''(x) = u(x)$ on $[0,1]$, with $v(0) = v(1) = 0$
  - $u(x)$ is the forcing term – where and how
    ”
  - $v(x)$ is the response – the shape that results

how hard you push

how much it deforms





Operator Learning vs PINNs

- PINNs are basically just the model of the month + PDE constrained training loss.
- In PINNs you specialize on a single problem and retrain every time!
- Operator Learning attempts to learn the underlying mapping of classes of problems.
- So if it's seemingly more useful why isn't it much more popular?

Operator Learning: Limitations

- Input is a function with ∞ DOFs (huge)
- Very hard to explore this space.
- So naturally they are very inaccurate.
- Relative error 10^{-4} vs 10^{-16} in classical solvers.
- G can also have complex structure
  - Shocks; boundary layers; bifurcations; anisotropy; variable domains

Newton–Kantorovich Method

- We want to solve $F(u,v)=0$ for nonlinear v
- Same idea as Newton's method but for functions:
  1. Start with a guess v_0
  2. Linearize F around v_n
  3. Solve for a correction δv_n
  4. Update $v_{n+1}=v_n+\delta v_n$
- A hard nonlinear problem is now a sequence of easy linear ones
- Each linear problem depends on the current estimate v_n, so it changes every iteration
- This is the standard algorithm for solving nonlinear PDEs numerically

CHONKNORIS

The paper.

This is the paper.

OPERATOR LEARNING AT MACHINE PRECISION

ARAS BACHO[A,1,2], ALEKSEI G. SOROKIN[B,2], XIANJIN YANG[A,2], THÉO BOURDAIS [A], EDOARDO CALVELLO[A], MATTHIEU DARCY[A], ALEXANDER HSU[C], BAMDAD HOSSEINI[C], HOUMAN OWHADI[A]

Abstract. Neural operator learning methods have garnered significant attention in scientific computing for their ability to approximate infinite-dimensional operators. However, increasing their complexity often fails to substantially improve their accuracy, leaving them on par with much simpler approaches such as kernel methods and more traditional reduced-order models. In this article, we set out to address this shortcoming and introduce CHONKNORIS (Cholesky Newton–Kantorovich Neural Operator Residual Iterative System), an operator learning paradigm that can achieve machine precision. CHONKNORIS draws on numerical analysis: many nonlinear forward and inverse PDE problems are solvable by Newton-type methods. Rather than regressing the solution operator itself, our method regresses the Cholesky factors of the elliptic operator associated with Tikhonov-regularized Newton–Kantorovich updates. The resulting unrolled iteration yields a neural architecture whose machine-precision behavior follows from achieving a contractive map, requiring far lower accuracy than end-to-end approximation of the solution operator. We benchmark CHONKNORIS on a range of nonlinear forward and inverse problems, including a nonlinear elliptic equation, Burgers' equation, a nonlinear Darcy flow problem, the Calderón problem, an inverse wave scattering problem, and a problem from seismic imaging. We also present theoretical guarantees for the convergence of CHONKNORIS in terms of the accuracy of the emulated Cholesky factors. Additionally, we introduce a foundation model variant, FONKNORIS (Foundation Newton–Kantorovich Neural Operator Residual Iterative System), which aggregates multiple pre-trained CHONKNORIS experts for diverse PDEs to emulate the solution map of a novel nonlinear PDE. Our FONKNORIS model is able to accurately solve unseen nonlinear PDEs such as the Klein–Gordon and Sine–Gordon equations.

Main Idea

Don't learn the full nonlinear map G, instead learn the solver used inside each Newton step

- If we try to learn G directly we're approximating a nonlinear function of ∞ variables. (hard)
- If we instead learn the inner solver we're approximating a class of matrices $A(u, v_n)^{-1}$ (way easier)
- Because we're learning the inner solve, the approximation just has to be good enough to yield a decent search direction

Main Idea

- The linear system in each step can be ill-conditioned (ie the numerics behave in a nasty way)
- Add a regularization term

$$\delta v_n = \arg\min_{\delta v} \left\| F + \frac{\delta F}{\delta v}[\delta v] \right\|^2 + \lambda \|\delta v\|^2$$

- This produces a damped update

$$\delta v_n = -\underbrace{\left(J^T J + \lambda I\right)^{-1}}_{Q(u,v_n)} J^T F(u, v_n)$$

where $J = \frac{\delta F}{\delta v}$ is the Jacobian

$Q(u,v)$ is the matrix we need each iteration and is what CHONKNORIS learns

Main Idea

- Learn $Q = (J^T J + \lambda I)^{-1}$ as a function of (u, v)
- Instead of learning Q directly, learn its Cholesky factors $\hat{R}$
- The update becomes

$$v_{n+1} = v_n - \alpha_n \hat{R}\hat{R}^T J^T F(u, v_n)$$

where α_n is a step size

Main Idea

- Learn $Q = (J^T J + \lambda I)^{-1}$ as a function of (u, v)
- Instead of learning Q directly, learn its Cholesky factors $\hat{R}$
- The update becomes

$$v_{n+1} = v_n - \alpha_n \hat{R}\hat{R}^T J^T F(u, v_n)$$

where α_n is a step size

Learned search direction

Nonlinear residual

Linear error minimizer

Training

- Generate a bunch of real Cholesky factors several iterations of the inner solve.
- Run any regression model to minimize

$$\sum \|\hat{R}_\theta - R_{\mathtt{true}}\|^2$$

- Because training data comes from the Newton solver's trajectory we greatly shrink the dimensions of the problem.
- Just learn how R changes as it the problem converges.

Inference

- Inference replaces the cost of recomputing and factorizing Q.
- Generate R and perform two TRSVs.
- Iterate until $\|F(u, v_n)\|$ is small enough.
- You can monitor convergence without knowing the true answer
- NOTE: operator learning methods generally don't let you do this, they're meant to be a one-shot method.

Inference

$$v_n \xrightarrow{\text{evaluate } F} r_n \xrightarrow{J^T} g_n \xrightarrow{\hat{R}\hat{R}^T} \delta v_n \xrightarrow{v_n + \delta v_n} v_{n+1}$$

- If you squint each iteration looks like a ResNet block
- Input v_n, compute a correction, add it back
- Key difference is that we don't explicitly have layers, we have iterations. And we don't have more expressivity, we have more accuracy (spoiler)

FONKNORIS

- Their foundational model.
- Main idea. the Jacobian of any second-order PDE can be written as

$$J = a(u,v)\,\partial_{xx} + b(u,v)\,\partial_x + c(u,v)$$

- Different PDEs just have different coefficient functions a,b,c
- FONKNORIS learns $\hat{R}$ as a function of a,b,c instead of u,v
- The model never sees the PDE itself just the structure of its linearization

FONKNORIS

- Train separate experts, one per PDE
- Combine them using a mixture-of-experts procedure
- Training data is roughly 10k samples per PDE at $N_x = 64$ grid points
- Test on new PDEs that we excluded from training (although the coefficients do fall within the range of training data)

Forward Problems

"We're good; they're bad."

Problem	GP	FNO	TNO	CHONKNORIS
Elliptic	$5e{-}6$	$1e{-}3$	$6e{-}3$	$9e{-}16$
Burgers'	$1e{-}1$	$9e{-}3$	$2e{-}2$	$5e{-}16$
Darcy flow	$2e{-}3$	$5e{-}3$	$4e{-}3$	$1e{-}15$

- Elliptic - 10 iterations to machine precision (exact solver needs 4)
- Darcy: needs more iterations for harder inputs
- 10 iters $= 1e{-}3$, 100 iters $= 1e{-}6$, 1000 iters $= 1e{-}16$
- Problems are all remarkably small

Forward Problems

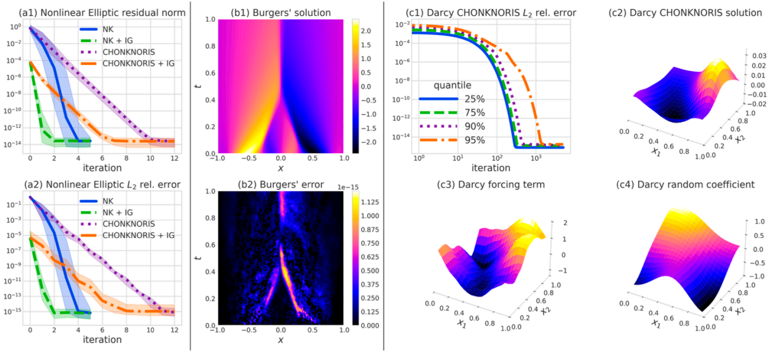


FIGURE 2. Forward problems. (a) Results for the nonlinear elliptic PDE problem. Quantiles of 10% − 90% are shown across test realizations. Our CHONKNORIS method is able to achieve machine precision accuracy in around 10 iterations. (b) Results for Burgers' equation. CHONKNORIS was able to achieve machine precision error in recovering the discretized solution which contained shocks. (c) Results for the Darcy flow PDE: (c1) shows that more challenging realizations require more CHONKNORIS iterations. (c4) shows a single realization of the random coefficient with the corresponding solution in (c2). (c3) shows the fixed forcing term.

Inverse Problems

"We're good; they're bad."

Problem	GP	CHONKNORIS
Impedance imaging	2e−2	3e−15
Wave scattering	2e−2	9e−13
Seismic imaging 5×5	2e−2	2e−14
Seismic imaging 10×10	6e−2	1e−3

- Inverse problems: given some measurements recover a bunch of hidden parameters
- These are much harder than forward problems (ill-posed, nonlocal)
- Seismic model fails at 10×10 seismic, the matrices become ill-conditioned

Generalizability

- Trained on nonlinear elliptic, Burgers', Darcy flow.
- Tested on Klein-Gordon, Sine-Gordon
- Achieves $1e{-}8$ - $1e{-}11$ relative error on test PDEs after 20 iters.
- 5–8 orders of magnitude better than any existing method.
- NOTE: conventional solvers converge 4-5.
- Why does this work? The authors aren't confident but it likely has something to do with the fact that inverting a linear operator is much simpler than one-shot operating learning.

RESULTS

Generalizability

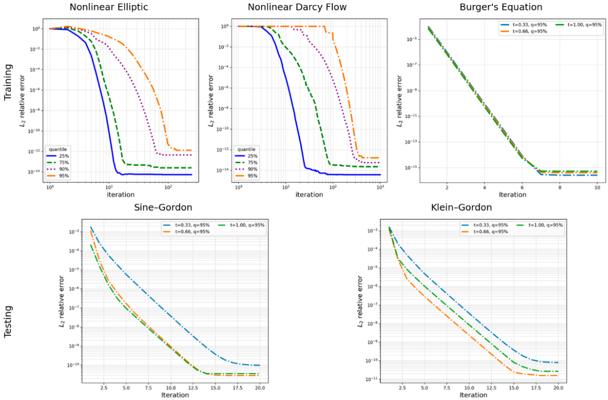


FIGURE 3. Quantiles of predictions of FONKNORIS for 100 realizations of initial conditions, external forces, and conductivities using a mixture of experts consisting of GPs for nonlinear elliptic, nonlinear darcy flow, and Burgers' equation and testing it for withheld Sine–Gordon and Klein–Gordon equations.

Convergence bounds

- Per-iteration "forcing term"

$$\theta_k \leq \frac{\lambda_k}{\lambda_k + \sigma_*^2} + M^2 \varepsilon_\lambda \hat{R}$$

- Left term - the error due to solving in a linearized iterative fashion .
- Right right - approximation error of the surrogate
- Authors prove the solution converges for $\theta_k < 1$.
- This means we don't need to learn the factors perfectly just less than the cost of the left term.
- (I am highly simplifying this bit)

That’s really it.