Physics-Informed Neural Networks (PINNs)

Bridging Physics and Deep Learning for ODE/PDE Problems

Marek Cieślar

Astronomical Observatory, University of Warsaw

October 15, 2025

Overview

- Motivation: Why Physics-Informed ML?
- What are PINNs?
- Advantages vs. traditional solvers
- Applications: Fluid dynamics, weather
- Applications: Astrophysics, self-gravity in gas, GW, Lane-Emden, cosmology-sth
- Challenges

Motivation – Bridging Physics and ML

- Traditional solvers: accurate but mesh- and timestep-heavy; struggle in high dimensions.
- Pure ML: flexible but can violate physics and overfit/out-of-domain fail.
- **PINNs:** embed governing equations (PDE/ODE) as soft constraints during training

What Are PINNs? – Core Concept

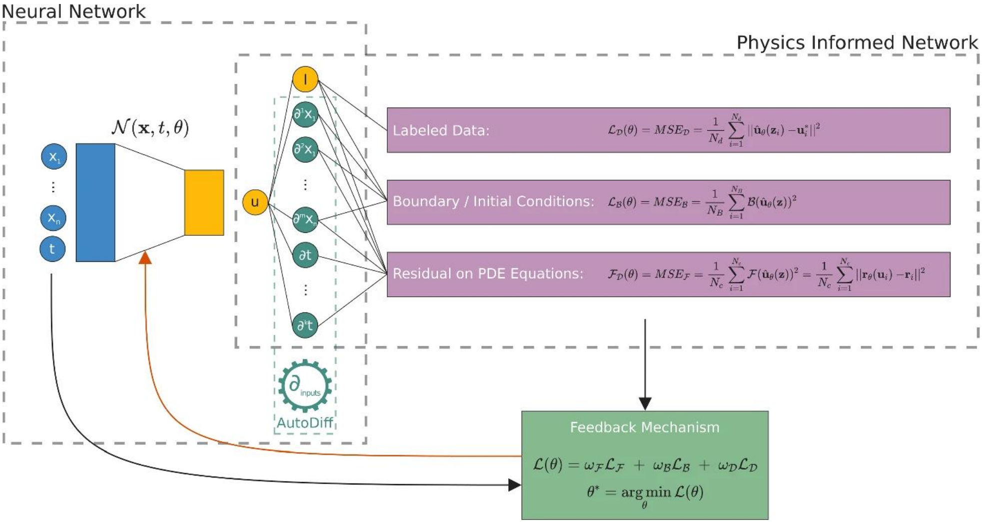


Figure 1: Cuomo et al. (2022)

How PINNs Are Trained

1. Define NN: inputs (coords/params), outputs (field(s)).
2. Form composite loss: $L = L_\mathrm{D} + L_\mathrm{F} + L_\mathrm{B}$.
3. Sample collocation points; compute residuals with AutoDiff.
4. Optimize (Adam $\rightarrow$ L-BFGS); monitor residuals/BCs
5. Enforce BCs softly (penalty) or hard (by design); validate.

Example: Pendulum - ML

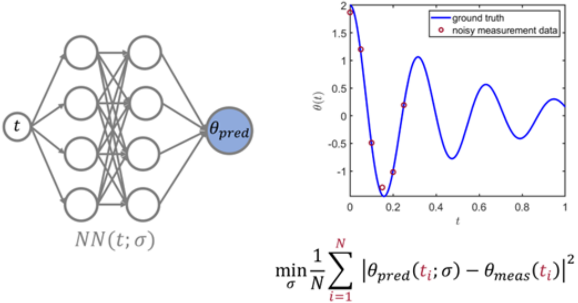


Figure 2: MathWorks: PINNs

Example: a damped pendulum - ML solution

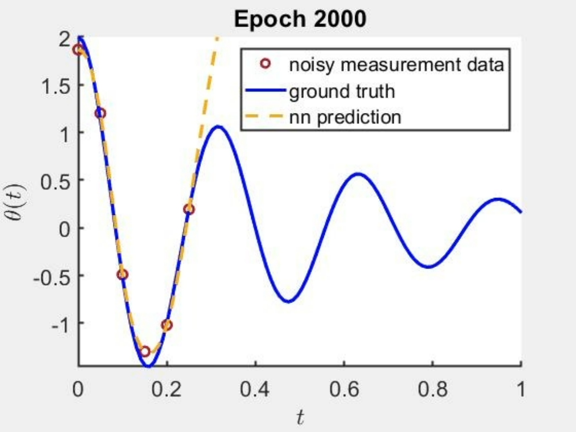
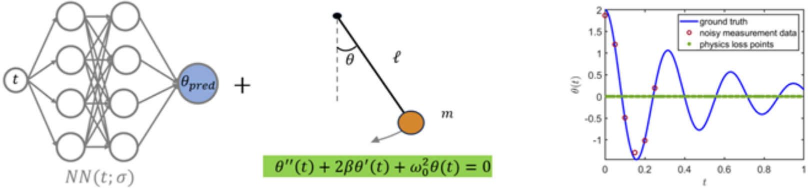


Figure 3: MathWorks: PINNs

Example: a damped pendulum - PINN



$$\min_{\sigma} \frac{1}{N}\sum_{i=1}^{N} \left|\theta_{pred}(t_i;\sigma) - \theta_{meas}(t_i)\right|^2$$

$$+\frac{1}{M}\sum_{j=1}^{M} \left|\frac{\partial^2}{\partial t^2}\theta_{pred}(t_j;\sigma) + 2\beta\frac{\partial}{\partial t}\theta_{pred}(t_j;\sigma) + \omega_0^2\theta_{pred}(t_j;\sigma)\right|^2$$

Figure 4: MathWorks: PINNs

Example: a damped pendulum - PINN solution

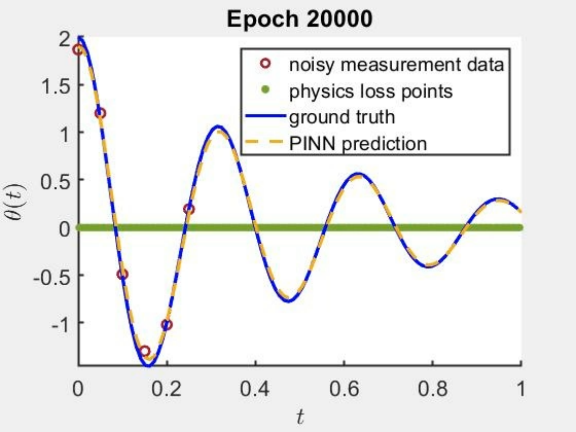


Figure 5: MathWorks: PINNs

Why PINNs? – Key Advantages

- Mesh-free and flexible; continuous solutions over domain.
- Physics-consistent; integrates sparse data.
- Unified forward and inverse framework.
- Often scales better in higher dimensions; substituting modeling.

Trade-off: ML vs. Numerical Methods vs. PINNs

	Purely Data-Driven Approaches	Traditional Numerical Methods	PINNs
Incorporate known physics	✗	✓	✓
Generalize well with limited or noisy training data	✗	✗	✓
Solve forward and inverse problems simultaneously	✓	✗	✓
Solve high-dimensional PDEs	✗	✗	✓
Enable fast "online" prediction	✓	✗	✓
Are mesh-free	✓	✗	✓
Have well-understood convergence theory	✗	✓	✗
Scale well to high-frequency and multiscale PDEs	✗	✓	✗

Figure 6: MathWorks: PINNs

Trade-off: ML vs. Numerical Methods vs. PINNs

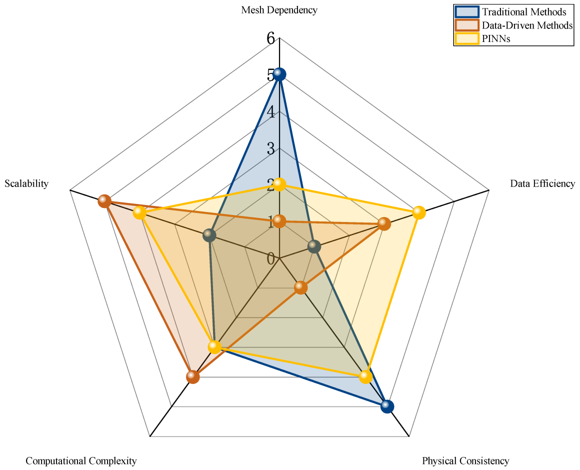


Figure 7: Ren et al. (2025)

Navier–Stokes (incompressible)

Governing PDEs (vector form):

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu\nabla^2\boldsymbol{u} + \boldsymbol{f}, \qquad \nabla\cdot\boldsymbol{u} = 0,$$

where $\boldsymbol{u}(\boldsymbol{x}, t)$ is velocity, $p(\boldsymbol{x}, t)$ pressure, ρ density (const.), ν kinematic viscosity, $\boldsymbol{f}$ body force.

Non-dimensional form: with characteristic (U, L),

$$\frac{\partial \boldsymbol{u}^*}{\partial t^*} + (\boldsymbol{u}^*\cdot\nabla^*)\boldsymbol{u}^* = -\nabla^* p^* + \frac{1}{\mathrm{Re}}\nabla^{*2}\boldsymbol{u}^* + \boldsymbol{f}^*, \qquad \nabla^*\cdot\boldsymbol{u}^* = 0,$$

with Reynolds number $\mathrm{Re} = \dfrac{UL}{\nu}$. where ν is the kinematic viscosity of the fluid $(\mathrm{m}^2/\mathrm{s})$. U Characteristic velocity scale — representative flow speed. L Characteristic length scale — representative geometric dimension.

- Low Re ($\lesssim 10^3$): viscous-dominated, laminar, smooth flow.
- High Re ($\gtrsim 10^4$): inertia-dominated, turbulent or unsteady flow.

Key Take-Aways: Raissi et al. (2020)

Goal:

- Introduced *Hidden Fluid Mechanics (HFM)*: a PINN-based framework that extracts hidden velocity and pressure fields from observations of a passive scalar (e.g. dye or smoke) under advection–diffusion + Navier–Stokes constraints.
- The method *encodes the NS + transport PDEs* into the network's loss, rather than learning from direct velocity/pressure labels.
- It is *agnostic to geometry, boundary conditions, or initial conditions* in its domain of interest — flexible in domain selection.

Demonstrated Capabilities:

- Successfully recovered velocity and pressure from sparse/noisy scalar concentration observations, even in irregular domains.
- Showed resilience to low resolution and significant noise in data, indicating practical utility in experimental settings.

Raissi et al. (2020)

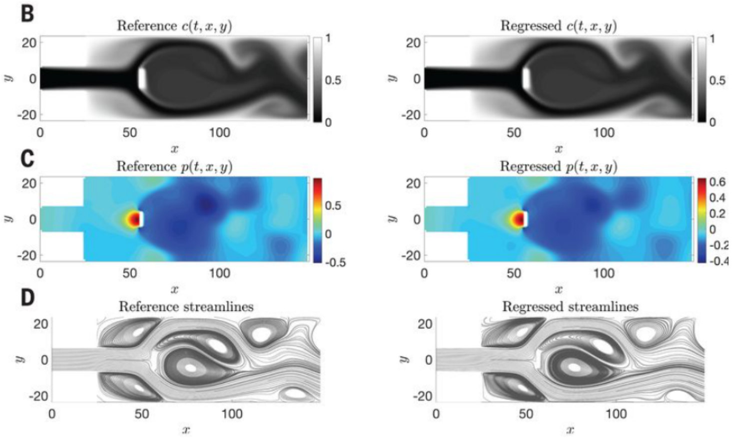


Figure 8: Flow around obstacle: PINN-reconstructed pressure/velocity from sparse tracer observations.

Weather Forecasting/Simulation - PINNs Review Kashinath et al. (2021)

Positive/negative side of PINNs:

- *forward* (solving PDEs) and *inverse / discovery* (parameter estimation, PDE learning) applications under one umbrella.
- *gradient pathologies*: imbalance between data loss and PDE residual loss can lead to poor convergence.

Applications and impact:

- PINNs applied across diverse domains: fluid mechanics, heat conduction, wave equations, inverse problems.
- Ability of PINNs to handle *multiphysics, multi-scale, and parameterized PDEs* with fewer data and mesh constraints.

Kashinath et al. (2021)

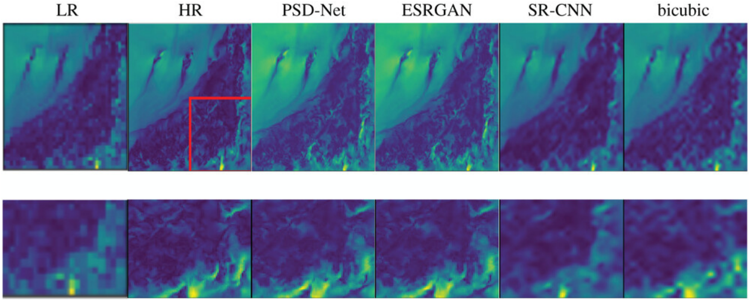


Figure 9: Images of the LR input, high-resolution ground truth (HR), and generated SR outputs from PSD-Net, ESRGAN, SRCNN and bicubic upsampling. Although ESRGAN performs poorly on PSNR, MSE and MAE, the generated images reveal that both PSD-Net and ESRGAN produce sharper images that have more realistic small-scale features and are less prone to artefacts.

Self-gracity in fluids: GRINN Auddy et al. (2023)

Problem:

- GRINN is a PINN designed for *self-gravitating hydrodynamics*, coupling fluid dynamics and gravity (Poisson equation) in a mesh-free framework.
- Targets simulation of gravitational instability and wave propagation in isothermal gas, across 1D, 2D, and 3D settings.

Performance:

- In the linear regime, GRINN matches analytic solutions to within $\simeq 1\%$ error; in the nonlinear regime, it stays within $\simeq 5\%$ compared to conventional grid codes.
- Demonstrated favorable scaling: in 3D, GRINN's runtime is *orders of magnitude lower* than a comparable finite difference code for similar accuracy.
- In lower dimensions (1D, 2D), GRINN is slower than conventional codes, but its performance advantage kicks in as dimensionality increases.

Auddy et al. (2023)

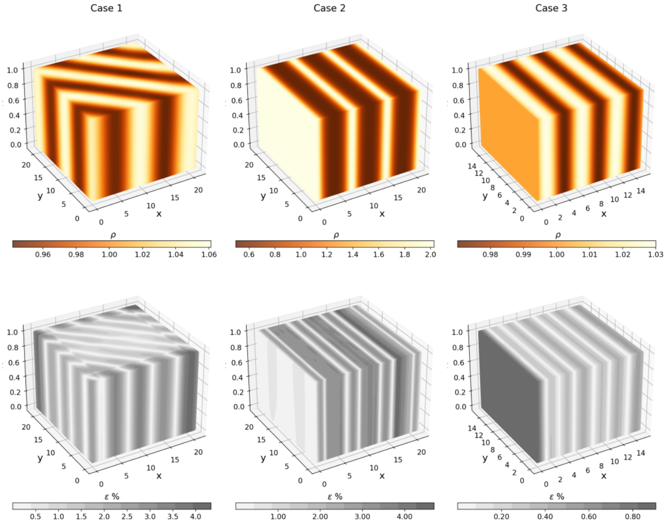


Figure 10: Top: GRINN density solutions for a 3D self-gravitating hydrodynamic system (three distinct cases). Bottom: The relative mismatch between the GRINN and standard FD (*Finite Difference* numerical method) solutions.

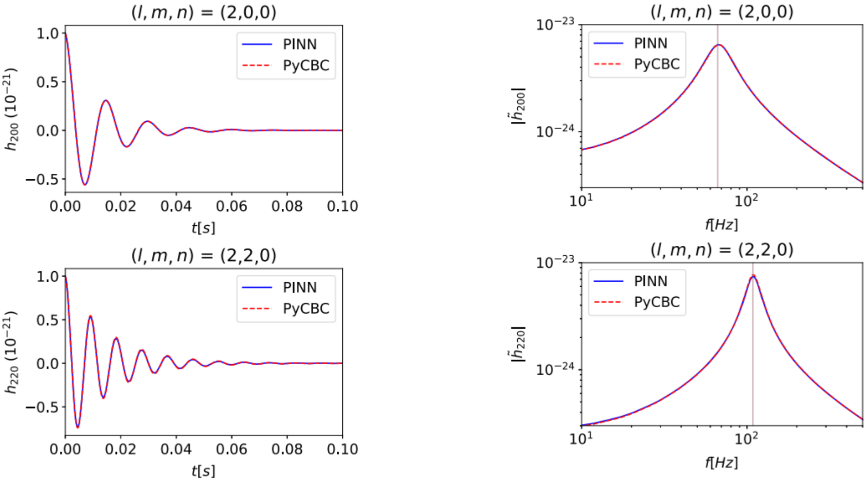
Astrophysics: Gravitational Waves from Kerr BH Luna et al. (2023)

- Teukolsky-PINN: able to compute Kerr black hole quasinormal modes (QNMs) with $\lesssim 1\%$ error vs. spectral benchmarks.
- Potential for fast parameter estimation in gravitational-wave pipelines.

What are QNMs?

- After a perturbation, a black hole *rings down* by emitting gravitational waves that damp with time.
- These are described by complex frequencies $\omega = \omega_R + i\,\omega_I$, where ω_R is oscillation frequency, and $\omega_I < 0$ is a decay (damping) rate.
- QNMs satisfy *boundary conditions*: purely ingoing at the event horizon, and purely outgoing at spatial infinity.

Luna et al. (2023)



Lane–Emden Equation and Polytropic Stars

Models a self-gravitating, spherically symmetric polytropic fluid ($P = K\rho^{1+\frac{1}{n}}$) in hydrostatic equilibrium. The *dimensionless form* of the equation:

$$\frac{1}{\xi^2}\frac{d}{d\xi}\left(\xi^2\frac{d\theta}{d\xi}\right) + \theta^n = 0,$$

with

$$\rho = \rho_c\,\theta^n, \quad P = K\rho_c^{1+1/n}\,\theta^{n+1}$$

where: $\xi = r/\alpha$ is the dimensionless radius, $\theta(\xi)$ is the dimensionless density (normalized so $\theta(0) = 1$.
With the boundary conditions:

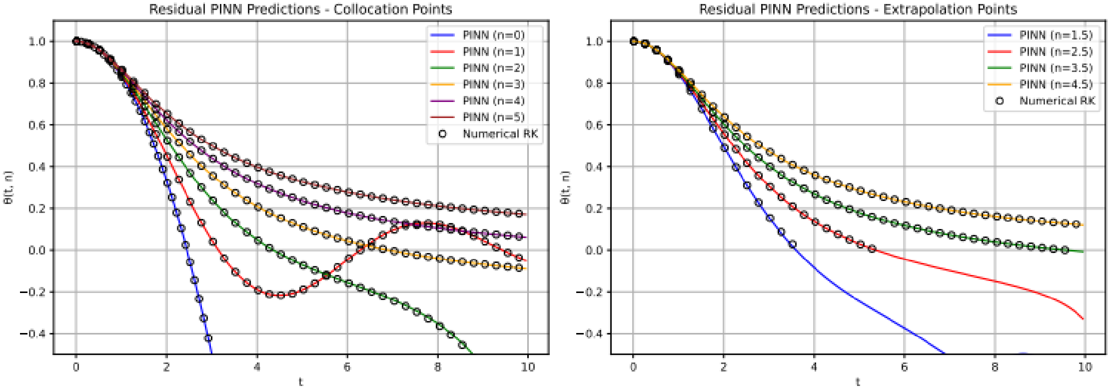
$$\theta(0) = 1, \quad \theta'(0) = 0.$$

Astrophysics: Stellar Structure (Lane–Emden) Mohuț and Popa (2025).

Insights:

- Radial coordinate and polytropic index as inputs to the PINN.
- The models not only fit the training indices n, but also extrapolate to unseen polytropic indices with high accuracy.
- Studies on width and depth: best performance often arises in moderate network complexity (e.g. 2–3 residual blocks with 64–256 hidden units) rather than extreme size.
- Embedding the parameter n as input lets a single PINN represent a family of ODEs, avoiding retraining for each index.
- Architectural design (residuals, gating, Fourier features) matters significantly for stability and representation of different regimes.

Mohuț and Popa (2025).



Cosmology-informed PINNs Verma et al. (2025).

Goal:

- To train a PINN surrogate to learn the normalized dark energy density $x_{\mathrm{de}}(z;\theta)$, taking redshift z and dark energy EoS parameters $\theta = (w_0, w_a)$ as inputs .
- To embedd directly into a *MCMC likelihood pipeline* using Pantheon+ supernova data to infer cosmological parameters.

Results:

- The surrogate reproduces the Hubble expansion $E(z)$ with *sub-percent errors* over most of the parameter–redshift domain (edges a bit worse).
- Distance modulus bias introduced by the surrogate remains below 0.1 magnitudes even up to $z \sim 2.5$, which is within the observational scatter of supernova data.
- Compared to direct ODE integration inside an MCMC, the surrogate becomes advantageous after about 4 independent runs (i.e. the break-even point of training cost vs repeated evaluations).

Figure 11: Mental health unit at the Royal Hospital in Edinburgh.

Limitations of PINNs

- Limited convergence theory
  - Theoretical guarantees for convergence of PINNs are still underdeveloped compared to classical numerical methods.
- Lack of unified training strategies
  - Training approaches (e.g., loss balancing, domain sampling) are often ad hoc and not standardized.
- Computational cost of calculating high-order derivatives
  - Automatic differentiation, while precise, becomes expensive and memory-intensive for high-order PDEs.
- Difficulty learning high-frequency and multiscale components of PDE solutions
  - PINNs often struggle to represent sharp gradients or fine-scale oscillations due to the smoothness bias of neural networks.

Frameworks for Physics-Informed Learning

- **PhysicsNeMo** — NVIDIA's open-source framework for scalable physics-AI models. Supports PINNs, neural operators, GNNs, and hybrid architectures; provides APIs to encode PDE constraints, geometry, and distributed training. (*large-scale industry-style modelling*)
- **DeepXDE** — A Python library for scientific ML and physics-informed learning (Lu et al. (2021)). Implements PINN, inverse PDE, fractional PDEs, operator learning; supports adaptive sampling, hard constraints, complex geometries. (*Academic research*)

Both frameworks provide all necessary building blocks (loss functions, domain management, autodiff, sampling strategies).

References

S. Auddy, R. Dey, N. J. Turner, and S. Basu. Grinn: A physics-informed neural network for self-gravitating hydrodynamic systems. *arXiv preprint arXiv:2308.08010*, 2023. URL https://arxiv.org/abs/2308.08010.

S. Cuomo, V. Schiano Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022. doi: 10.1007/s10915-022-01939-z. URL https://doi.org/10.1007/s10915-022-01939-z.

K. Kashinath, M. Mustafa, A. Albert, J.-L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H. A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, and n. Prabhat. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194): 20200093, 2021. doi: 10.1098/rsta.2020.0093. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2020.0093.

L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021. doi: 10.1137/19M1274067.

R. Luna, E. Cruz-Osorio, B. Bonga, H. Olivares, Z. Younsi, et al. Solving the teukolsky equation with physics-informed neural networks. *Physical Review D*, 107 (6):064025, 2023. doi: 10.1103/PhysRevD.107.064025.

A.-I. Mohuț and C.-A. Popa. Generalization-capable pinns for the lane–emden equation: Residual and stellarnet approaches. *Applied Sciences*, 15(18):10035, 2025. doi: 10.3390/app151810035.

M. Raissi, A. Yazdani, and G. E. Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481): 1026–1030, 2020. doi: 10.1126/science.aaw4741.

Z. Ren, S. Zhou, D. Liu, and Q. Liu. Physics-informed neural networks: A review of methodological evolution, theoretical foundations, and interdisciplinary frontiers. *Applied Sciences*, 15(14):8092, 2025. doi: 10.3390/app15148092.

A. Verma, J. Doe, and R. Smith. Cosmology-informed neural networks to infer dark energy equation-of-state. arXiv preprint arXiv:2508.12032, 2025. URL https://arxiv.org/abs/2508.12032. Preprint.