GlueNN

Gluing patchwise analytic solutions with neural networks

ML4HEP-V4 (2026 - TIFR Mumbai)

Group Members: Yashasvi, Abhinav Kumar, Astha Tiwari,
Purnath Unnikrishnan, Kerman Zaveri,
Tagtshen Tamang, Eeshita

Mentor: Dr. Partha Konar, Dr. Akanksha Bhardwaj

Motivation

- Traditional NNs are like Black-boxes.
- PINNs incorporate governing physics equations into the learning process.
- But we often know the answer in patches.
- So why make the network re-derive known physics?
- Result: better interpretability with limited solution.

Method Overview

Goal: Solve D[f(x)] = 0 continuously across the entire domain.

Region 1 ($x < a$)	**The Unknown Region** ($a \leq x \leq b$)	**Region 2** ($x > b$)
Known Solutions	**The Intermediate Gap**	**Known Solutions**
$f(x) = f_1(x, c_1)$ (Asymptotic Analytic)	No closed-form solution exists. Bridge the barrier!	$f(x) = f_2(x, c_2)$ (Asymptotic Analytic)

$x = a$ $\qquad$ $a \leq x \leq b$ $\qquad$ $x = b$

Analytic Path 1 | **Neural Optimized Transition** | **Analytic Path 2**

The idea: promote constants to functions

Instead of learning the solution directly, GlueNN learns how the integration constants vary

Classically

$$f_j(x, c_j)$$

c_j is constant.

GlueNN

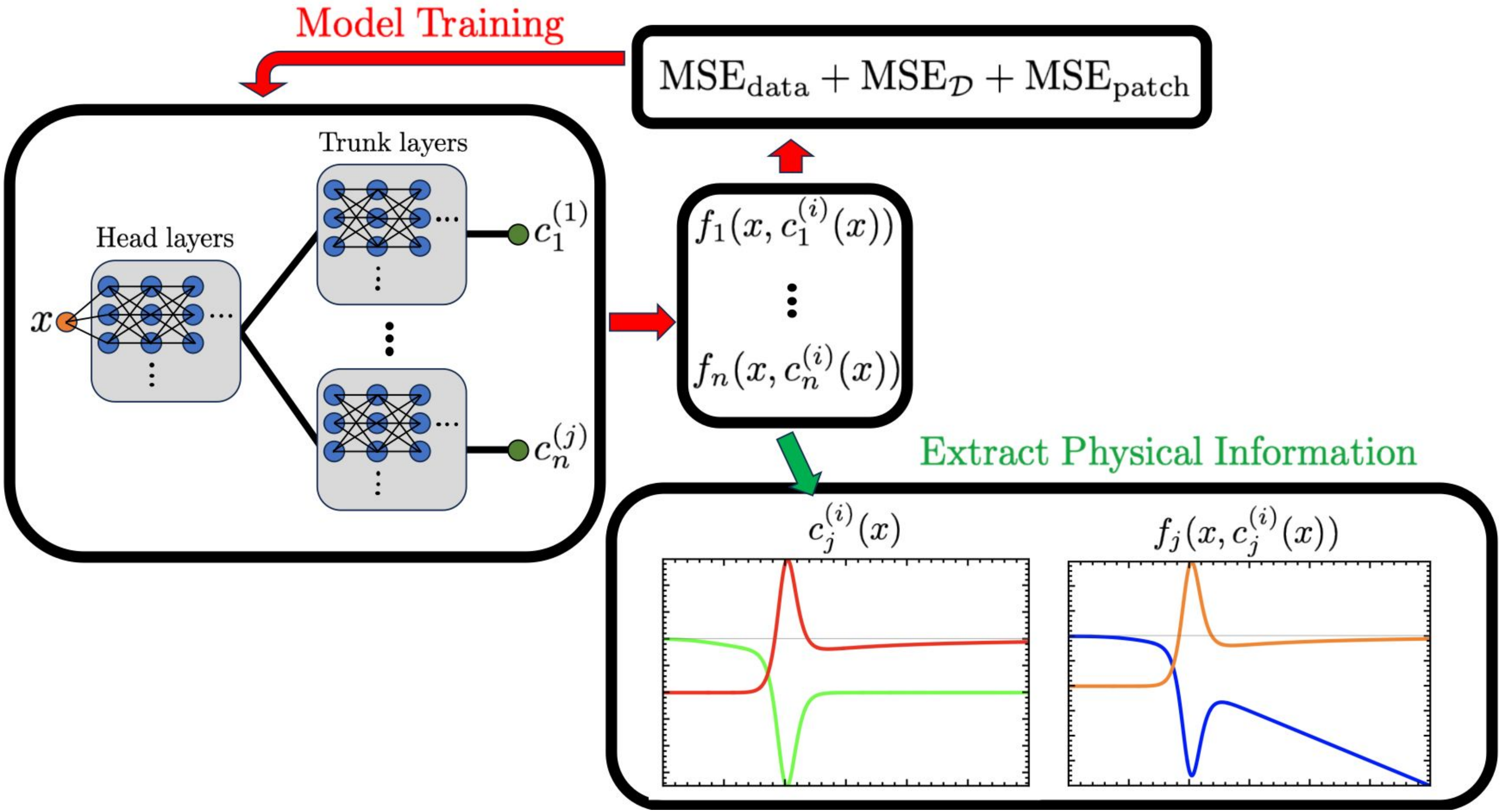
$$f_j(x, c_j(x))$$

$c_j(x)$ is network output.

Global ansatz: $y(x) = \sum_j f_j(x, c_j(x))$

- Coefficients converge to true analytical constants deep inside each patch.
- Coefficients are suppressed toward zero outside their respective patches.
- The transition profile is learned directly from the differential equation.

Architecture



https://arxiv.org/abs/2601.05889

The Loss Function

$$\mathbf{L} = \mathbf{MSE}_{\mathbf{data}} + \mathbf{MSE}_{\mathbf{D}} + \mathbf{MSE}_{\mathbf{patch}}$$

$$\mathrm{MSE}_{\mathrm{data}} = \frac{\lambda_a}{N_a} \sum_{i,j,\, x \in \{x_\alpha\}} \left| f_j\left(x, c_j^{(i)}(x)\right) - h(x) \right|^2$$

- Matches the known solution at sampled data points.

$$\mathrm{MSE}_D = \frac{\lambda_b}{N_b} \sum_{i,j,\, x \in \{x_\beta\}} \left| \frac{D\left[f_j\left(x, c_j^{(i)}(x)\right)\right]}{\mathcal{N}\left[f_j\left(x, c_j^{(i)}(x)\right)\right]} \right|^2$$

- Minimizes the differential equation residual.

$$\mathrm{MSE}_{\mathrm{patch}} = \frac{\lambda_c}{N_c} \sum_{i,\, x \in \{x_\gamma\}} \left| f_j\left(x, c_j^{(i)}(x)\right) \right|^2$$

- Suppresses inactive analytic patches.

Example: Quantum Tunneling

$$\mathcal{D}[\psi(x)] = -\frac{\hbar}{2m}\frac{d^2\psi}{dx^2} + V(x)\psi - E\psi = 0$$

$$V(x) = \begin{cases} V_0, & |x| \leq \dfrac{d}{2}, \\ 0, & \text{otherwise} \end{cases}$$

For |x| $\gg$ d/2, the wave function reduces to free particle equation,

$$\psi(x) = Ae^{ikx} + Be^{-ikx} \qquad ; \quad k^2 = 2mE/\hbar^2$$

For |x| < d/2, the wave function becomes,

$$\psi(x) = Ce^{\kappa x} + De^{-\kappa x} \qquad ; \quad \kappa^2 = 2m(V_0 - E)/\hbar^2$$

For simplicity, the wave functions is decomposed into its real and imaginary components, which can be solved independently.

$$Re[\psi(x)] = c_1^{(1)} \cos(kx) + c_1^{(2)} \sin(kx) + c_2^{(1)} e^{\kappa x} + c_2^{(2)} e^{-\kappa x}$$

$$Im[\psi(x)] = c_3^{(1)} \cos(kx) + c_3^{(2)} \sin(kx) + c_4^{(1)} e^{\kappa x} + c_4^{(2)} e^{-\kappa x}$$

The loss functions for real part are,

$$\mathrm{MSE}_{\mathrm{data}} = \frac{\lambda_a}{N_a} \sum_{x \in \{x_\alpha\}} |\mathrm{Re}[\psi(x)] - h(a)|^2 ,$$

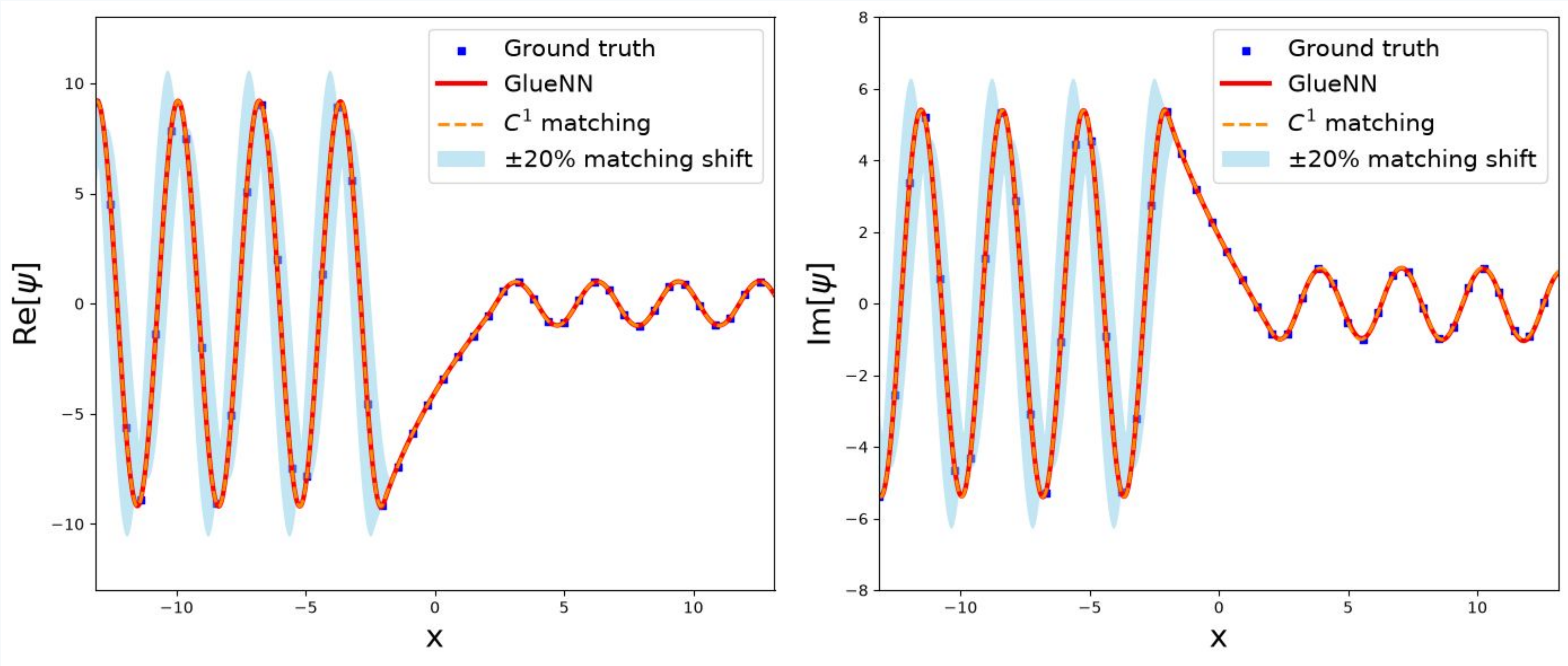
$$\mathrm{MSE}_{\mathcal{D}} = \frac{\lambda_b}{N_b} \sum_{x \in \{x_\beta\}} \left| \frac{\mathcal{D}[\mathrm{Re}[\psi(x)]]}{|\mathrm{Re}[\psi(x)]| + \epsilon} \right|^2 ,$$

$$\mathrm{MSE}_{\mathrm{patch}}^{(1)} = \frac{\lambda_c}{N_c} \sum_{x \in \{x_\gamma\}} \left| c_1^{(1)} \cos(kx) + c_1^{(2)} \sin(kx) \right|^2 ,$$

$$\mathrm{MSE}_{\mathrm{patch}}^{(2)} = \frac{\lambda_d}{N_d} \sum_{x \in \{x_\delta\}} \left| c_2^{(1)} e^{\kappa x} + c_2^{(2)} e^{-\kappa x} \right|^2 .$$

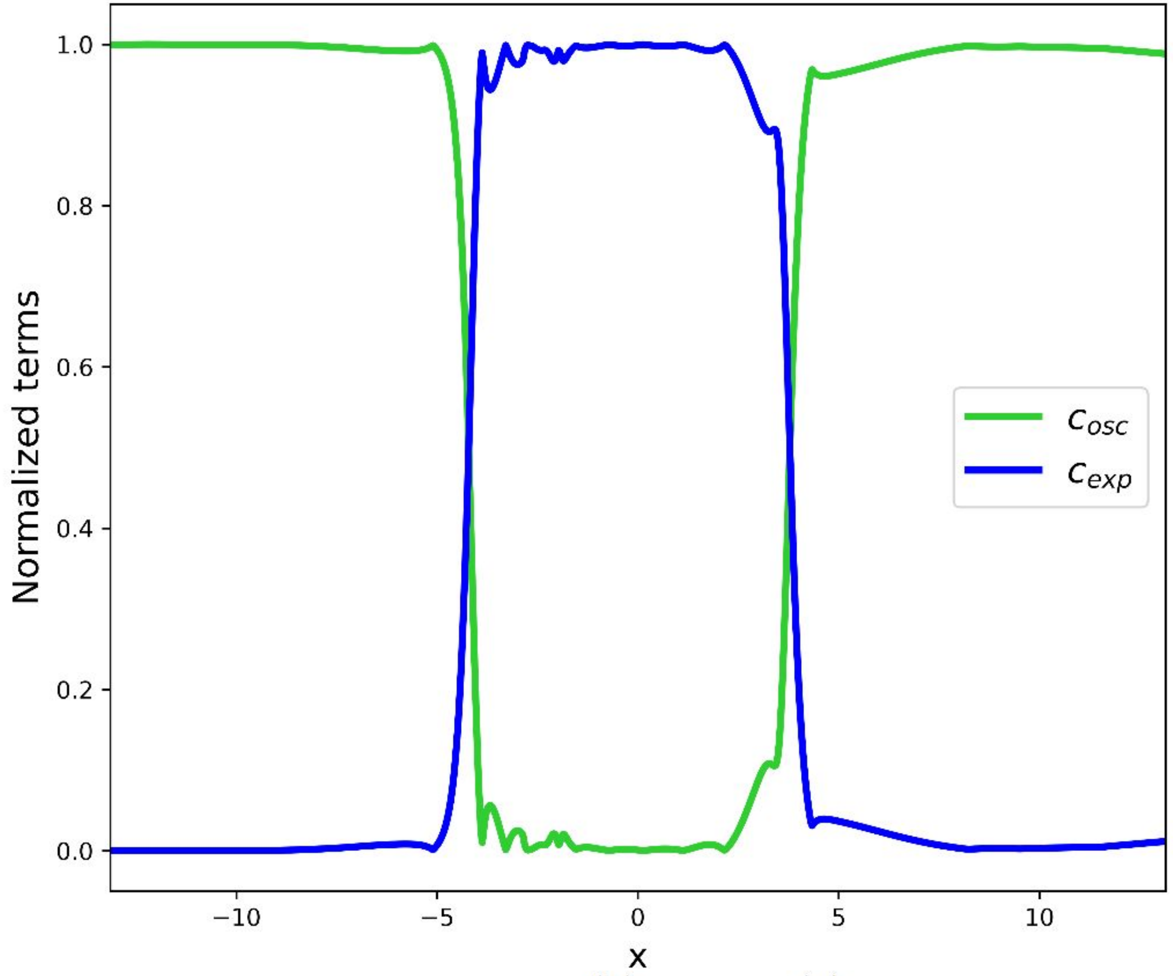
The loss for the imaginary part can be obtain in the same way by replacing {c1,c2,Re[ψ(x)]} with {c3,c4,Im[ψ(x)]} respectively.

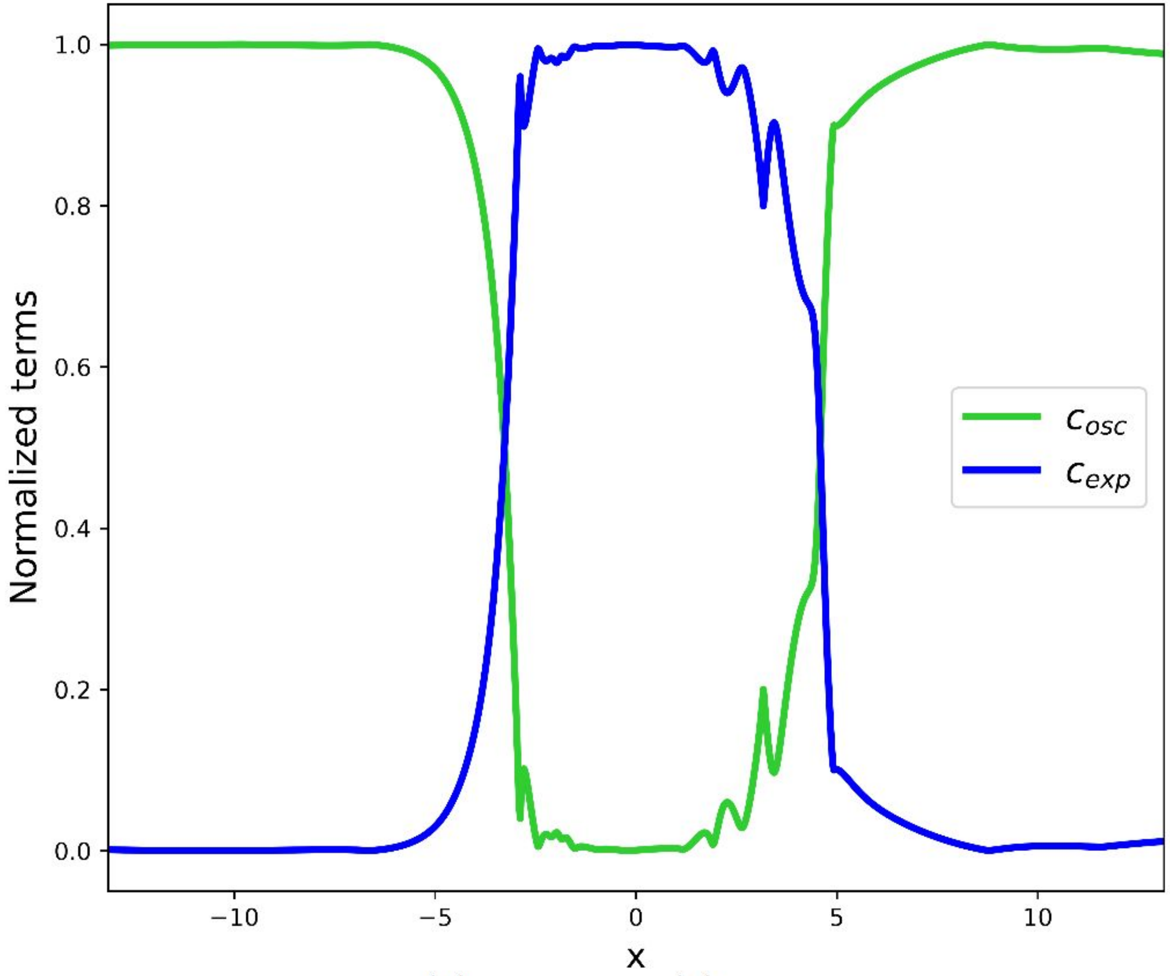
Results



Ground truth
GlueNN
C^1 matching
±20% matching shift
Re[ψ]
x
Ground truth
GlueNN
C^1 matching
±20% matching shift
Im[ψ]
x

Results





$$\tilde{c}_{\mathrm{osc},r} = \frac{(c_1^{(1)})^2 + (c_1^{(2)})^2}{\mathcal{N}_r},$$

$$\tilde{c}_{\mathrm{osc},i} = \frac{(c_3^{(1)})^2 + (c_3^{(2)})^2}{\mathcal{N}_i},$$

$$\tilde{c}_{\mathrm{exp},r} = \frac{|c_2^{(1)}|e^{\kappa x} + |c_2^{(2)}|e^{-\kappa x}}{\mathcal{N}_r},$$

$$\tilde{c}_{\mathrm{exp},i} = \frac{|c_4^{(1)}|e^{\kappa x} + |c_4^{(2)}|e^{-\kappa x}}{\mathcal{N}_i},$$

Another Example: Chemical Reaction

$$\mathcal{D}[Y(x)] = \frac{dY}{dx} + \eta x^{-2}(Y^2 - Y_{\rm eq}^2) = 0,$$

$$Y_{\rm eq}(x) = 0.145\, x^{3/2} e^{-x}$$

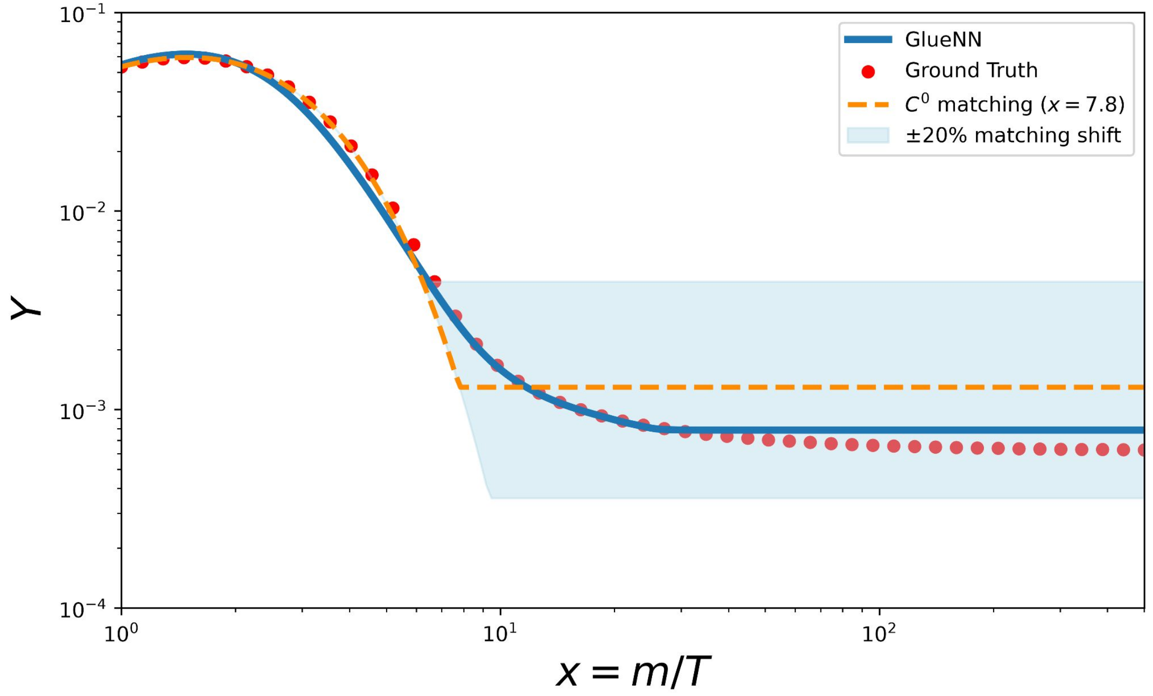
Parameterized Solution

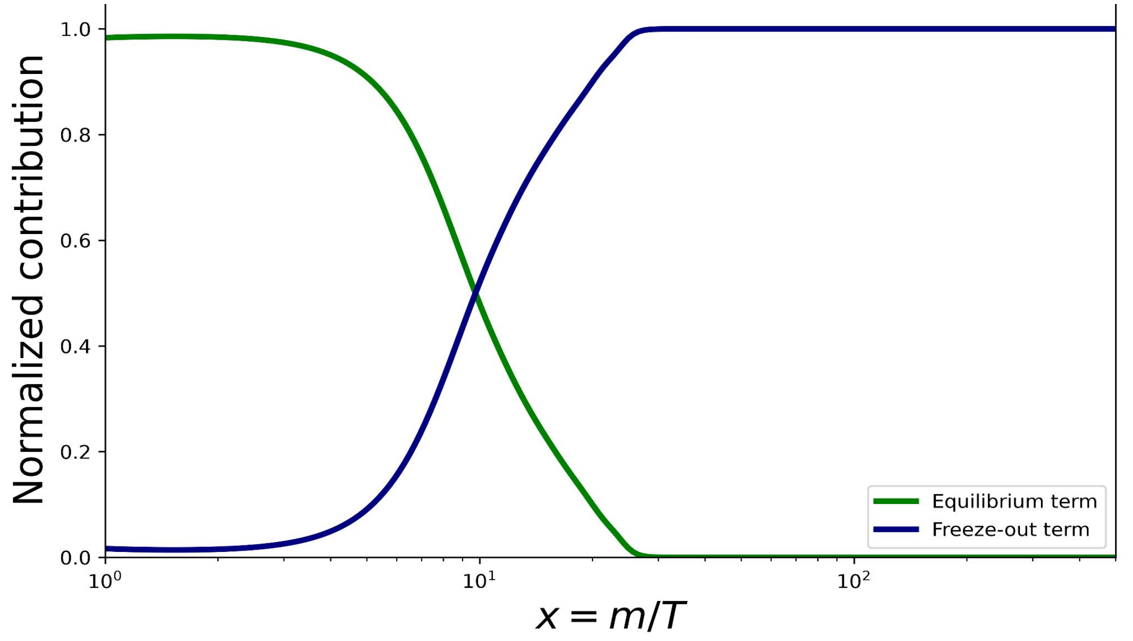
$$y(x) = e^{c_1^{(1)}(x)} x^{3/2} e^{-x} + e^{c_2^{(1)}(x)}$$

Loss Function

$$\mathrm{MSE}_{\rm data} = \frac{\lambda_a}{N_a} \sum_{x\in\{x_\alpha\}} |y(x) - h(x)|^2,$$

$$\mathrm{MSE}_{\mathcal{D}} = \frac{\lambda_b}{N_b} \sum_{x\in\{x_\beta\}} \left|\frac{\mathcal{D}[y(x)]}{y(x)}\right|^2,$$





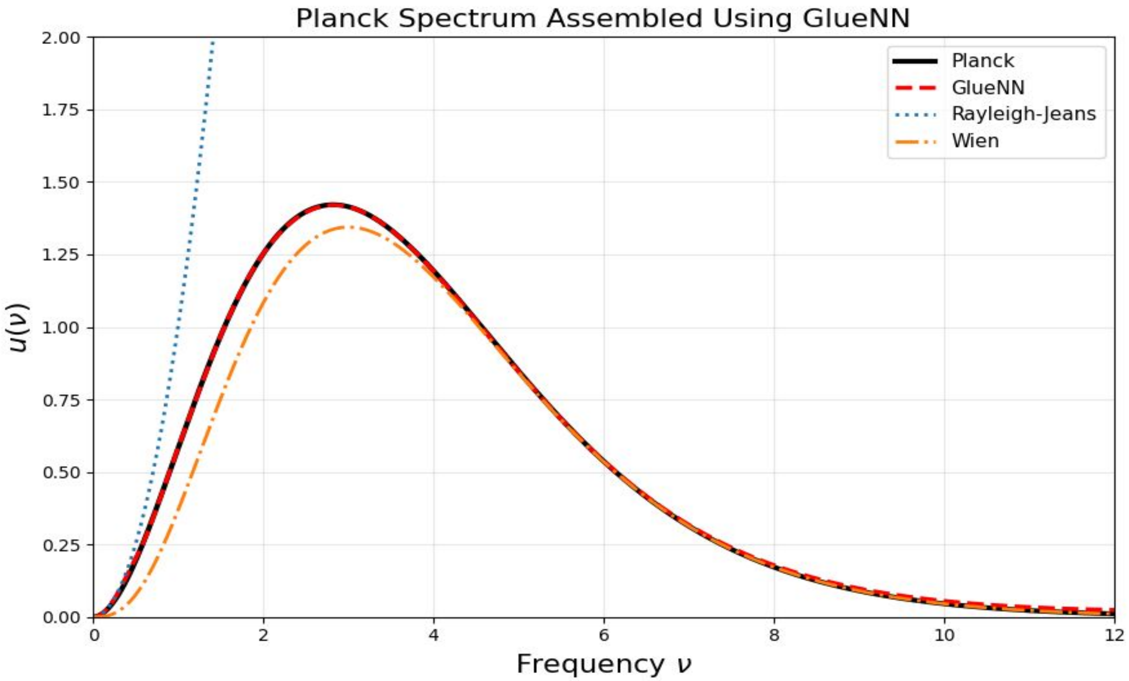
Another Example: Blackbody radiation

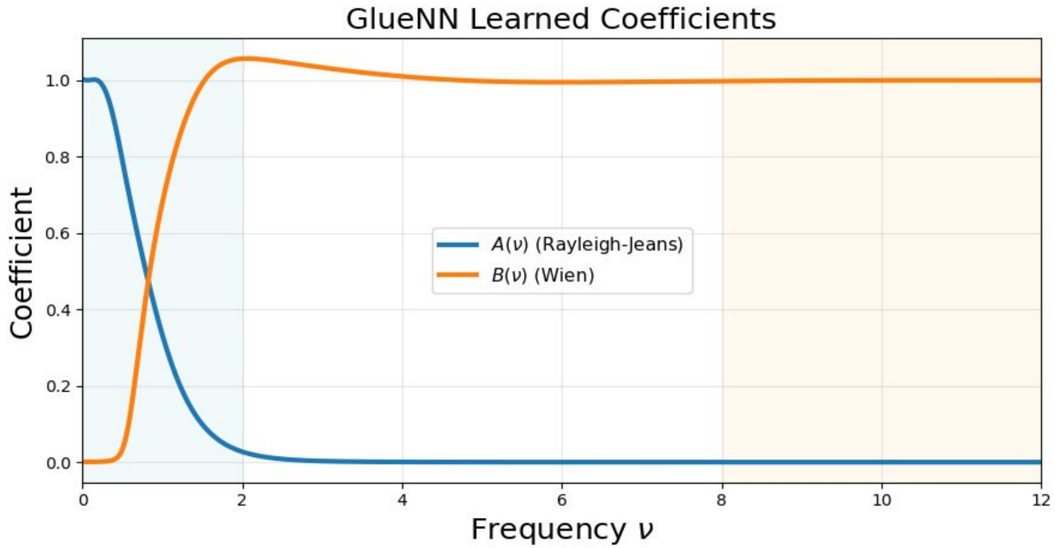
$$u_{\mathrm{RJ}}(\nu) = \frac{\nu^2}{\beta},$$

$$u_{\mathrm{Wien}}(\nu) = \nu^3 e^{-\beta\nu},$$

$$u_{\mathrm{GlueNN}}(\nu) = A(\nu)\,u_{\mathrm{RJ}}(\nu) + B(\nu)\,u_{\mathrm{Wien}}(\nu),$$

$$Loss:\ \mathcal{L} = \lambda_a L_{\mathrm{data}} + \lambda_b L_{\mathrm{boundary}} + \lambda_c L_{\mathrm{smooth}} + \lambda_d L_{\mathrm{coeff}}.$$





Key takeaways

- Traditional NN, learns everything from data.
- PINN: make sure the physics information remains intact
- GlueNN: Using the known information about the asymptotic forms and use NN to patch them

Backup: Boated potential