Born-Preconditioned Learning for Iterative Neural PDE Solvers

Juntao Wang[1,3] **Jiwei Jia**[1,3*] **Xinliang Liu**[2,3*]
[1]School of Mathematics, Jilin University, Changchun 130012, China
[2]Ocean University of China, Qingdao 266100, China
[3]Shenzhen Loop Area Institute, Shenzhen 518038, China

Abstract

Simulating high-frequency waves in spatially varying media remains challenging for both classical iterative methods and neural PDE solvers. We propose Neural Preconditioned Born Series (NPBS), a framework for learning iterative neural PDE solvers. Our guiding idea is to use a simple reference preconditioner to help learn a more effective neural preconditioner. Building on the Convergent Born Series (CBS), NPBS feeds a preconditioned residual to a neural network that predicts a solution update. We train the network to reduce the preconditioned residual after each update. The same reference preconditioner is used for the network input and the loss, giving a metric-matched training objective. Experiments on heterogeneous Helmholtz problems show reduced iteration counts across neural architectures and larger gains at higher wavenumbers. Within a Krylov solver, NPBS achieves the lowest wall-clock time in the evaluated solver comparison. The framework also extends naturally to three-dimensional problems. The same metric-matched formulation also improves convergence on convection–diffusion–reaction systems and Newton linear systems for nonlinear PDEs, indicating that NPBS provides a general framework for learning neural preconditioners. Code is available at `https://github.com/wangjt23/NeuralPreconditionedBornSeries`.

1 Introduction

Efficient solution of wave equations in heterogeneous media is central to a broad range of engineering and scientific applications, from seismic inversion and medical ultrasound imaging to electromagnetic scattering and photonic design. Many of these problems reduce to the heterogeneous Helmholtz equation, whose high-wavenumber and high-contrast regimes remain challenging for both classical and learning-based solvers [17, 54]. The main obstacles are well known: standard discretizations suffer from pollution errors [2], multilevel methods face coarse-grid dispersion mismatch [48], and indefiniteness can undermine relaxation and coarse correction [16]. Classical approaches address these issues through shifted-Laplacian preconditioning [15, 14], wave-ray multigrid [5], Convergent Born Series (CBS) [39], and domain decomposition [9, 18]. Despite these advances, robust and efficient solution of heterogeneous Helmholtz problems remains difficult.

Learning-based PDE solvers have developed rapidly in recent years. Physics-informed neural networks (PINNs) [41] encode governing equations and boundary conditions directly into the training objective, but often require a separate optimization for each new instance, limiting cross-instance generalization and amortized efficiency [49]. Neural operators address this limitation by learning mappings between function spaces, allowing one model to solve families of PDEs parameterized by coefficients, sources, or geometries. Representative works include DeepONet [37], FNO [34, 35], and MgNO [21], among many others [27, 6, 7, 38]. While neural operators greatly improve scalability,

*Corresponding authors: `jiajiwei@jlu.edu.cn` (Jiwei Jia); `xinliang.liu@ouc.edu.cn` (Xinliang Liu).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

end-to-end operator learning remains difficult for highly oscillatory solutions. Because of the spectral bias of neural networks [40, 36], high-frequency components are typically much harder to learn accurately than low-frequency ones. For high-frequency Helmholtz equations, this makes one-shot wavefield prediction particularly challenging [22].

A promising alternative is to embed a learned module inside an iterative solver, using the network as a residual-to-correction map rather than as a direct solution predictor [26, 43, 56]. Recent learned iterative solvers have shown that this strategy can reduce iteration counts or wall-clock time relative to classical baselines [52], making this direction especially promising for challenging Helmholtz problems. Our work follows this direction, but asks a more specific question: what residual should a correction network receive, and what residual norm should its training loss reduce?

We answer this question by proposing *Neural Preconditioned Born Series* (NPBS), a learned iterative preconditioning framework inspired by the CBS. For Helmholtz problems, the Born residual is exactly the residual of a shifted-Laplacian left-preconditioned system. This equivalence identifies the Born-preconditioned residual coordinates as a natural state space for learned iterative correction. NPBS builds on this view by replacing the diagonal CBS correction with a neural residual-to-correction map acting on Born-preconditioned residuals. To train this map, we minimize the relative norm of the Born-preconditioned residual after one correction. This residual forms the network input at the next NPBS step. Let G_η denote the Green operator of the damped reference operator. We introduce the residual metric induced by G_η:

$$\|\boldsymbol{r}\|_{R_\eta} := \|G_\eta \boldsymbol{r}\|_2, \qquad R_\eta := G_\eta^* G_\eta, \tag{1}$$

and derive a metric-matched Born-series objective $\mathcal{L}_{\mathrm{bs}}^{R_\eta}$ that aligns training and inference within the same preconditioned geometry. This metric reweights residual modes through the reference inverse, whose response can be highly nonuniform near resonance [16, 14]. The same construction applies to convection–diffusion–reaction (CDR) systems and Newton linear systems arising from nonlinear PDEs [20, 29].

Our contributions are summarized as follows:

1. **A learned Born-preconditioned iterative framework.** We introduce NPBS, which lifts the diagonal CBS preconditioner to an architecture-agnostic learned residual-to-correction map operating on Born-preconditioned residuals.
2. **A metric-matched training principle induced by the reference Green operator.** By identifying Born residuals with shifted-Laplacian left-preconditioned residuals, we derive the induced metric $R_\eta = G_\eta^* G_\eta$ and the objective $\mathcal{L}_{\mathrm{bs}}^{R_\eta}$, making residual-metric matching the principle that aligns training with preconditioned inference.
3. **Robustness to ill-conditioning and algorithmic generalization.** The two-dimensional FNO study shows iteration reductions of $1.2\times$–$1.9\times$ over direct learning as wavenumber increases, at 0.7% additional training cost. Further tests cover four additional neural backbones, three-dimensional Helmholtz problems, CDR systems, and Newton linear systems. FGMRES with NPBS attains the lowest solve time among the evaluated two-dimensional classical and learned solvers.

The paper is organized as follows. Section 2 positions NPBS relative to neural preconditioners and Born-inspired neural solvers. Section 3 establishes the algebraic equivalence between Born-series residuals and shifted-Laplacian left preconditioning. Section 4 introduces the NPBS iteration and derives the metric-matched training objective. Section 5 reports experiments on heterogeneous Helmholtz benchmarks across difficulty levels and backbone architectures, with extensions to CDR and nonlinear PDEs.

2 Related Work

Neural preconditioners and learned iterative solvers. Recent learning-based solvers increasingly use neural networks as components of an iterative numerical method rather than as standalone solution predictors. Examples include learned multigrid and meta-multigrid schemes [19, 8], neural-operator-accelerated Krylov methods [43, 26], hybrids of neural operators and relaxation methods [56, 28, 23, 10], neural preconditioned Newton-type solvers [29], and preconditioners based

on learned Green's functions [50]. Learned solvers have also been developed for Helmholtz equations [46, 30], including multigrid-augmented deep preconditioners [1, 31], Wave-ADR-NS [11], and MGCFNN [52]. Training objectives have also been tailored to solver properties, using an A-energy error loss in FCG-NO [43], data-dependent losses for learned SPD preconditioners in PCG [32], or scale-invariant losses for sparse approximate inverses in CG [53]. Other approaches combine condition and residual losses [33] or optimize residual–subspace angles through differentiable FGMRES [13]. NPBS uses the same reference inverse to precondition the network input and the residual measured by the training loss.

Born-inspired neural solvers. The classical Born series/Lippmann–Schwinger formulations express wave scattering through a reference Green operator and a medium perturbation; CBS [39] modifies this construction to obtain a convergent fixed-point method for heterogeneous Helmholtz problems. Recent learned Born-series methods, such as Strong Scattering Neural Operator ($\mathrm{S}^2\mathrm{NO}$) [55] and Learned Born Series (LBS) [47], use Born-style unrolling to build forward wavefield simulators for scattering media. Their central use of the Born structure is therefore as a learned finite-depth surrogate for the wavefield map. NPBS instead uses Born-preconditioned residuals to iteratively update the solution, with a residual-based stopping criterion, and can also serve as a preconditioner in FGMRES.

3 From Born Series to Left Preconditioning

3.1 Operator Splitting and Integral Equation

We consider the Helmholtz equation for wave scattering in an unbounded heterogeneous medium, subject to the Sommerfeld radiation condition:

$$\begin{aligned} \mathcal{L}u(x) &:= -\Delta u(x) - k(x)^2 u(x) = f(x), \quad x \in \mathbb{R}^d, \\ \lim_{|x|\to\infty} |x|^{\frac{d-1}{2}} &\left(\frac{\partial u}{\partial |x|} - ik_0 u \right) = 0, \end{aligned} \tag{2}$$

where u is the wavefield, f the source, d the spatial dimension, and $k(x) = \omega/c(x)$ the spatially varying wavenumber (ω: angular frequency, $c(x)$: speed of sound). The constant k_0 is the background wavenumber ($k(x) \to k_0$ as $|x| \to \infty$).

For computation, the unbounded domain is truncated to a bounded domain Ω, and absorbing boundary layers—sponge layers [25] or PML [3, 46]—are incorporated so that the resulting bounded-domain operator (still denoted $\mathcal{L}$) suppresses spurious reflections. After discretization (finite differences or FFT-based pseudospectral methods), we obtain the linear system

$$A\boldsymbol{u} = \boldsymbol{f}, \tag{3}$$

where A is the discrete Helmholtz operator (including absorbing-layer terms), $\boldsymbol{u}$ the discrete unknown, and $\boldsymbol{f}$ the discrete source. Introduce a damped reference operator

$$\mathcal{L}_\eta := -\Delta - (k_0^2 + i\eta), \qquad \eta > 0, \tag{4}$$

used only for preconditioning/training coordinates. The reference is chosen to be transform-diagonalizable (periodic FFT or Dirichlet/Neumann DST/DCT) for fast application; any mismatch between this reference and the physical operator, including absorbing-layer terms, is absorbed into the perturbation

$$\mathcal{V}_\eta := \mathcal{L}_\eta - \mathcal{L}. \tag{5}$$

Let g_η be the Green kernel for the *continuous* reference operator $\mathcal{L}_\eta$ under the chosen boundary condition on Ω. Define the associated continuous Green operator

$$(\mathcal{G}_\eta q)(x) := \int_\Omega g_\eta(x, y)\, q(y)\, \mathrm{d}y, \qquad q \in L^2(\Omega), \tag{6}$$

so that $\mathcal{G}_\eta = \mathcal{L}_\eta^{-1}$ on the admissible function class.

Discrete implementation. In the discrete setting, let L_η denote the matrix obtained by discretizing $\mathcal{L}_\eta$ (e.g., via finite differences or Fourier pseudospectral methods; see Appendix D for details), and define the discrete Green operator $G_\eta := L_\eta^{-1}$. For transform-diagonalizable references (periodic FFT or Dirichlet/Neumann DST/DCT), G_η can be applied in $\mathcal{O}(N \log N)$ operations (Appendix C).

Operator splitting and Lippmann–Schwinger form. With $\mathcal{V}_\eta := \mathcal{L}_\eta - \mathcal{L}$, the heterogeneous Helmholtz operator splits as

$$\mathcal{L} = \mathcal{L}_\eta - \mathcal{V}_\eta. \tag{7}$$

For the coefficient-only Helmholtz split, $\mathcal{V}_\eta$ reduces to multiplication by $k(x)^2 - k_0^2 - i\eta$.

In the continuous setting, $\mathcal{L}u = f$ implies

$$\mathcal{L}_\eta u = \mathcal{V}_\eta u + f, \qquad u = \mathcal{G}_\eta(\mathcal{V}_\eta u + f), \tag{8}$$

equivalently,

$$(\mathcal{I} - \mathcal{G}_\eta \mathcal{V}_\eta)u = \mathcal{G}_\eta f. \tag{9}$$

In the discrete setting, starting from $A\boldsymbol{u} = \boldsymbol{f}$ and the discrete split $A = L_\eta - V_\eta$, we obtain

$$(I - G_\eta V_\eta)\boldsymbol{u} = G_\eta \boldsymbol{f}. \tag{10}$$

Equation (10) is the preconditioned operator equation used by CBS [39], whose fixed-point iteration is

$$\boldsymbol{u}^{m+1} = \boldsymbol{u}^m + \gamma(G_\eta(V_\eta \boldsymbol{u}^m + \boldsymbol{f}) - \boldsymbol{u}^m)\,, \tag{11}$$

where γ is a spatially varying diagonal preconditioner derived from the perturbation V_η.

3.2 Equivalence to Shifted-Laplacian Left Preconditioning

We state the algebraic equivalence that defines the preconditioned system underlying NPBS.

Proposition 3.1 (Algebraic equivalence to left preconditioning)**.** *Let the discrete Helmholtz operator admit the split $A = L_\eta - V_\eta$, and assume L_η is invertible so that $G_\eta := L_\eta^{-1}$ is well-defined. Then*

$$I - G_\eta V_\eta = G_\eta A = L_\eta^{-1} A. \tag{12}$$

Hence the Lippmann–Schwinger equation (10) *is algebraically identical to the preconditioned system*

$$L_\eta^{-1} A\, \boldsymbol{u} = L_\eta^{-1} \boldsymbol{f}. \tag{13}$$

Proof. Since $G_\eta L_\eta = I$, we have $I - G_\eta V_\eta = G_\eta(L_\eta - V_\eta) = G_\eta A$. □

The identity (12) identifies $G_\eta \boldsymbol{r}$ as the residual of the left-preconditioned system, with the induced metric $R_\eta = G_\eta^* G_\eta$.

For a residual $\boldsymbol{r}$, the correction equation $A\delta\boldsymbol{u} = \boldsymbol{r}$ is equivalently $(G_\eta A)\delta\boldsymbol{u} = G_\eta \boldsymbol{r}$. When A is nonsingular, this identifies $(G_\eta A)^{-1}$ as the ideal map from the Born-preconditioned residual to the solution correction.

4 Neural Preconditioned Born Series Framework

A common learned linear-solver baseline uses a neural correction map $\mathcal{M}_\theta$ driven by the *unpreconditioned* residual. (Since the learned map depends on the PDE coefficients a, it should formally be written $\mathcal{M}_\theta(\,\cdot\,; a)$; for brevity we write $\mathcal{M}_\theta(\,\cdot\,)$ below. The specific instantiation is given in Appendix E.) Given the discrete system $A\boldsymbol{u} = \boldsymbol{f}$, define the residual $\boldsymbol{r}^m := \boldsymbol{f} - A\boldsymbol{u}^m$ and iterate

$$\boldsymbol{u}^{m+1} = \boldsymbol{u}^m + \mathcal{M}_\theta(\boldsymbol{r}^m) = \boldsymbol{u}^m + \mathcal{M}_\theta(\boldsymbol{f} - A\boldsymbol{u}^m), \tag{14}$$

which operates on the unpreconditioned residual.

A standard training objective for $\mathcal{M}_\theta$ is to minimize the relative residual of the learned inverse action on random probes. Let $\mathcal{D}$ denote a probe distribution (e.g., $\mathcal{N}(0, I)$ or residual-replay samples). Then:

$$\mathcal{L}_{\mathrm{dir}}(\theta) = \mathbb{E}_{\boldsymbol{r}\sim\mathcal{D}} \frac{\|A\,\mathcal{M}_\theta(\boldsymbol{r}) - \boldsymbol{r}\|_2}{\|\boldsymbol{r}\|_2}. \tag{15}$$

This objective is simple and widely used [11, 52], but it optimizes the residual in *unpreconditioned* Euclidean geometry; for indefinite Helmholtz operators, near-resonant modes can yield stiff gradients and slow learning.

4.1 Neural Preconditioned Born Series (NPBS) iteration

Residual-to-correction map. Let $\boldsymbol{r}^m = \boldsymbol{f} - A\boldsymbol{u}^m$ and $\boldsymbol{r}^m_{\mathrm{bs}} := G_\eta \boldsymbol{r}^m$. CBS and NPBS use the same Born-preconditioned residual, but map it to a solution correction differently:

$$\delta\boldsymbol{u}^m_{\mathrm{CBS}} = \gamma(G_\eta \boldsymbol{r}^m), \qquad \delta\boldsymbol{u}^m_{\mathrm{NPBS}} = \mathcal{M}_\theta(G_\eta \boldsymbol{r}^m).$$

NPBS retains the reference inverse G_η and replaces the diagonal CBS correction γ with a learned map. The network receives $\boldsymbol{r}^m_{\mathrm{bs}}$ and returns the correction added to $\boldsymbol{u}^m$:

$$\boldsymbol{u}^{m+1} = \boldsymbol{u}^m + \mathcal{M}_\theta(G_\eta(\boldsymbol{f} - A\boldsymbol{u}^m)). \tag{16}$$

If $\mathcal{M}_\theta = \gamma$, this reduces to CBS ((11)). If A is nonsingular and $\mathcal{M}_\theta = (G_\eta A)^{-1}$, one step gives the exact solution.

Efficient integral-form implementation. The identity (12) gives $G_\eta(\boldsymbol{f} - A\boldsymbol{u}^m) = G_\eta(V_\eta \boldsymbol{u}^m + \boldsymbol{f}) - \boldsymbol{u}^m$, so the same update can be evaluated as

$$\boldsymbol{u}^{m+1} = \boldsymbol{u}^m + \mathcal{M}_\theta(G_\eta(V_\eta \boldsymbol{u}^m + \boldsymbol{f}) - \boldsymbol{u}^m). \tag{17}$$

Here $V_\eta \boldsymbol{u}^m$ is pointwise multiplication and applying G_η requires one transform pair (e.g., FFT/IFFT). Computing $A\boldsymbol{u}^m$ may require an additional application of the discrete Laplacian (e.g., an extra FFT in pseudospectral settings), hence we implement the efficient integral form (17) in practice.

4.2 Metric-matched training within the NPBS framework

With the input $G_\eta \boldsymbol{r}$ fixed, we compare two ways to measure the residual after a learned correction. The first retains the Euclidean norm of the physical residual:

$$\mathcal{L}^{\ell_2}_{\mathrm{bs}}(\theta) = \mathbb{E}_{\boldsymbol{r}\sim\mathcal{D}} \frac{\|A\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - \boldsymbol{r}\|_2}{\|\boldsymbol{r}\|_2}. \tag{18}$$

For a physical residual $\boldsymbol{r}$, let $\boldsymbol{z} = G_\eta \boldsymbol{r}$. After the correction $\mathcal{M}_\theta(\boldsymbol{z})$, the physical and Born-preconditioned residuals are

$$\boldsymbol{r}^+ = \boldsymbol{r} - A\mathcal{M}_\theta(\boldsymbol{z}), \qquad \boldsymbol{z}^+ = G_\eta \boldsymbol{r}^+ = \boldsymbol{z} - G_\eta A\mathcal{M}_\theta(\boldsymbol{z}).$$

The per-sample term in (18) is $\|\boldsymbol{r}^+\|_2/\|\boldsymbol{r}\|_2$, the relative physical residual after the correction. The next NPBS step receives $\boldsymbol{z}^+$. We therefore train the map to minimize the average ratio $\|\boldsymbol{z}^+\|_2/\|\boldsymbol{z}\|_2$. Using (12), this gives the metric-matched objective:

$$\mathcal{L}^{R_\eta}_{\mathrm{bs}}(\theta) = \mathbb{E}_{\boldsymbol{r}\sim\mathcal{D}} \frac{\|(I - G_\eta V_\eta)\,\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - G_\eta \boldsymbol{r}\|_2}{\|G_\eta \boldsymbol{r}\|_2}. \tag{19}$$

Proposition 4.1 (Equivalent weighted-residual formulation)**.** *To make the geometry underlying* (19) *explicit, define the metric induced by the reference Green operator:*

$$R_\eta := G^*_\eta G_\eta, \qquad \|\boldsymbol{x}\|_{R_\eta} := \|G_\eta \boldsymbol{x}\|_2, \qquad \langle \boldsymbol{x}, \boldsymbol{y}\rangle_{R_\eta} := \langle \boldsymbol{x}, R_\eta \boldsymbol{y}\rangle_2. \tag{20}$$

Then (19) *is equivalently*

$$\mathcal{L}^{R_\eta}_{\mathrm{bs}}(\theta) = \mathbb{E}_{\boldsymbol{r}\sim\mathcal{D}} \frac{\|A\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - \boldsymbol{r}\|_{R_\eta}}{\|\boldsymbol{r}\|_{R_\eta}}. \tag{21}$$

The proof is given in Appendix A.1.

The losses $\mathcal{L}^{\ell_2}_{\mathrm{bs}}$ and $\mathcal{L}^{R_\eta}_{\mathrm{bs}}$ use the same input $G_\eta \boldsymbol{r}$ and the same physical residual $A\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - \boldsymbol{r}$; only the norm differs. Thus, relative to (15), the progression is: unpreconditioned probes with Euclidean norm ($\mathcal{L}_{\mathrm{dir}}$), then preconditioned probes with Euclidean norm ($\mathcal{L}^{\ell_2}_{\mathrm{bs}}$), and finally preconditioned probes with the R_η-metric ($\mathcal{L}^{R_\eta}_{\mathrm{bs}}$). A detailed Fourier-domain interpretation of the metric reweighting is given in Appendix A.2.

A sufficient condition for convergence and a perturbation analysis are provided in Appendix A.3.

The CDR and Newton derivations, transform implementations, and network details are provided in Appendices B–E.

5 Experiments

We organize the evaluation around the heterogeneous Helmholtz equation. (i) we isolate the effects of iteration format and training loss across dataset difficulty and wavenumber regimes. (ii) we test whether the resulting gains persist across multiple neural-operator backbones. (iii) we compare stationary CBS and NPBS, and evaluate classical and learned preconditioners inside FGMRES. (iv) we test the same iteration and loss design on three-dimensional Helmholtz problems. (v) we evaluate on convection–diffusion–reaction (CDR) systems and Newton linear systems for nonlinear PDEs to demonstrate generality beyond Helmholtz. All methods stop when $\|\boldsymbol{f} - A\boldsymbol{u}^m\|_2 / \|\boldsymbol{f}\|_2 \leq \mathrm{rTol}$, with the tolerance specified for each experiment. The primary learned preconditioner uses an FNO backbone (Appendix E); dataset splits, training hyperparameters, and compute resources are detailed in Appendices G and H.

5.1 Heterogeneous Helmholtz Equation

For the two-dimensional Helmholtz experiments, we select two datasets from OpenFWI [12] with progressive difficulty: CurveVel-A (smooth curved velocity profiles) and CurveFault-B (sharp contrasts and fault structures); see Figure 5 for representative samples. Both use a 256×256 grid padded to 320×320 with 32-cell PML boundaries on each side, at 12 points per wavelength (ppw). Full dataset and discretization details are given in Appendix F.1.

5.1.1 Iteration format and loss comparison

We isolate the effect of the network iteration format and the choice of training loss on solver efficiency. The evaluation compares three configurations under a unified stopping rule (rTol $= 10^{-6}$): (i) Direct iteration trained with $\mathcal{L}_{\mathrm{dir}}$, (ii) NPBS iteration trained with $\mathcal{L}_{\mathrm{bs}}^{\ell_2}$, and (iii) NPBS iteration trained with the metric-matched objective $\mathcal{L}_{\mathrm{bs}}^{R_\eta}$. Comparing (i) and (ii) isolates the change in residual input, while comparing (ii) and (iii) isolates the choice of residual metric.

Table 1: Iteration format and loss comparison on OpenFWI datasets. Lower ppw corresponds to higher wavenumber. Ratio: Direct+$\mathcal{L}_{\mathrm{dir}}$ iteration count divided by NPBS+$\mathcal{L}_{\mathrm{bs}}^{R_\eta}$ iteration count.

Dataset	ppw	Direct + $\mathcal{L}_{\mathrm{dir}}$		NPBS + $\mathcal{L}_{\mathrm{bs}}^{\ell_2}$		NPBS + $\mathcal{L}_{\mathrm{bs}}^{R_\eta}$		Ratio
		Avg. Iters	Avg. Time	Avg. Iters	Avg. Time	Avg. Iters	Avg. Time	
CurveVel-A	12	491	0.499 s	466	0.470 s	**300**	**0.301 s**	1.6
CurveFault-B	24	531	0.537 s	517	0.537 s	**458**	**0.403 s**	1.2
	20	616	0.642 s	539	0.550 s	**475**	**0.465 s**	1.3
	16	895	0.914 s	842	0.848 s	**538**	**0.544 s**	1.7
	12	1051	1.030 s	859	0.838 s	**556**	**0.571 s**	1.9

Table 1 reports the effect of iteration scheme and training loss across OpenFWI datasets and wavenumber regimes. NPBS + $\mathcal{L}_{\mathrm{bs}}^{R_\eta}$ consistently requires the fewest iterations and achieves the lowest runtime; its advantage over Direct + $\mathcal{L}_{\mathrm{dir}}$ grows both with stronger heterogeneity, from $1.6\times$ on CurveVel-A to $1.9\times$ on CurveFault-B, and with increasing wavenumber, from $1.2\times$ to $1.9\times$ as ppw decreases from 24 to 12. These trends indicate that metric matching is most beneficial in harder Helmholtz regimes, while the runtime gains and the small 0.7% training overhead (Appendix I) corroborate the efficient integral-form design in §4.1.

5.1.2 Cross-backbone robustness of iteration format and loss

We compare all three formulations across five backbones: FNO [34], U-Net [42], MgNO [21], Encoder–solver based on [31], and MGCFNN [52]. The FNO row reproduces the CurveFault-B, PPW 12 results from Table 1. Implementation details and baseline adaptations are described in Appendix G.1.

Table 2 shows two robust trends. First, for every backbone, NPBS+$\mathcal{L}_{\mathrm{bs}}^{R_\eta}$ yields the lowest iteration count. Second, solve-time ranking differs from iteration ranking: although MGCFNN attains the

fewest iterations, MgNO gives the lowest average solve time, reflecting its lightweight design and compile-friendly implementation.

Table 2: Cross-backbone comparison on CurveFault-B with representative learning-based baselines.

Backbone	Direct + $\mathcal{L}_{\text{dir}}$		NPBS + $\mathcal{L}_{\text{bs}}^{\ell_2}$		NPBS + $\mathcal{L}_{\text{bs}}^{R_\eta}$	
	Avg. Iters	Avg. Time	Avg. Iters	Avg. Time	Avg. Iters	Avg. Time
FNO	1051	1.030 s	859	0.838 s	556	0.571 s
U-Net	853	0.433 s	496	0.254 s	399	0.198 s
MgNO	342	0.268 s	141	0.123 s	123	**0.107 s**
Encoder–solver	468	0.562 s	404	0.455 s	321	0.350 s
MGCFNN	69	0.336 s	66	0.334 s	**48**	0.235 s

5.1.3 Comparison with classical and learned preconditioners

Following the cross-backbone study, we select MgNO for its lowest average solve time among the evaluated neural architectures. On CurveFault-B, we compare stationary CBS [39] and NPBS, together with FGMRES [44] using CBS, shifted-Laplacian multigrid [15] (SL), the reference inverse G_η, or learned MgNO corrections. All FGMRES runs use restart length 50.

Table 3 shows that stationary NPBS reduces the average iteration count by $9.1\times$ relative to CBS, although its solve time is longer (0.107 vs. 0.086 s). FGMRES with metric-matched NPBS achieves the lowest reported solve time, with 37 iterations and 0.078 s. FGMRES with directly trained MgNO requires 47 iterations and 0.116 s. Using G_η alone requires 894 iterations and 2.821 s, so the reference inverse alone does not explain the learned preconditioner's performance. Figure 1 depicts a representative CurveFault-B sample with its wavefield solution and convergence profiles.

Table 3: Classical and learned solvers on CurveFault-B. All learned corrections use MgNO.

Method	Avg. Iters	Avg. Time
CBS	1124	0.086 s
FGMRES+CBS	547	1.820 s
FGMRES+SL	781	4.312 s
FGMRES+G_η	894	2.821 s
NPBS (MgNO)	123	0.107 s
FGMRES+MgNO	47	0.116 s
FGMRES+NPBS (MgNO)	**37**	**0.078** s

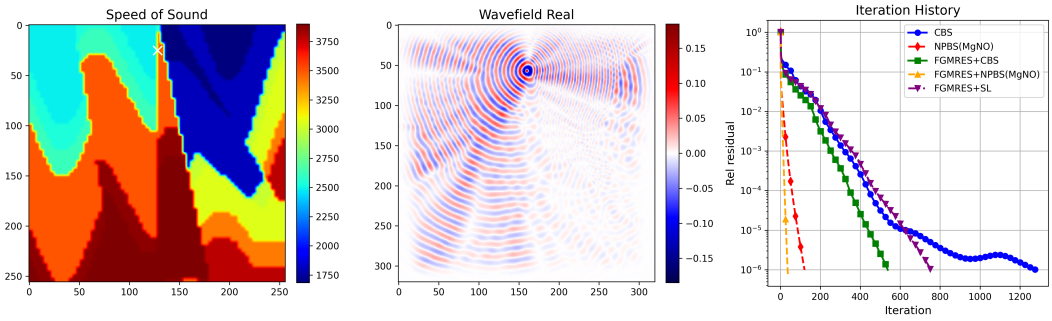

Figure 1: Iteration comparison of different strategies on a representative instance from the OpenFWI CurveFault-B dataset.

5.1.4 Three-dimensional Helmholtz experiments

NPBS extends naturally to three dimensions with the same iteration and loss formulation. We evaluate the three formulations on OpenFWI Kimberlina-V1 using a three-dimensional MgNO backbone on

192^3 grids at PPW 6. Both NPBS variants improve over direct residual learning, with the metric-matched formulation achieving the fewest iterations and lowest solve time (Table 4). Figure 2 illustrates a representative test case and its convergence histories. Experimental details are provided in Appendix G.1.

Table 4: Three-dimensional Kimberlina-V1 Helmholtz results with MgNO.

	Direct + $\mathcal{L}_{\text{dir}}$	NPBS + $\mathcal{L}_{\text{bs}}^{\ell_2}$	NPBS + $\mathcal{L}_{\text{bs}}^{R_\eta}$
Avg. Iters	445	68	**41**
Avg. Time	39.1 s	5.69 s	**3.60 s**

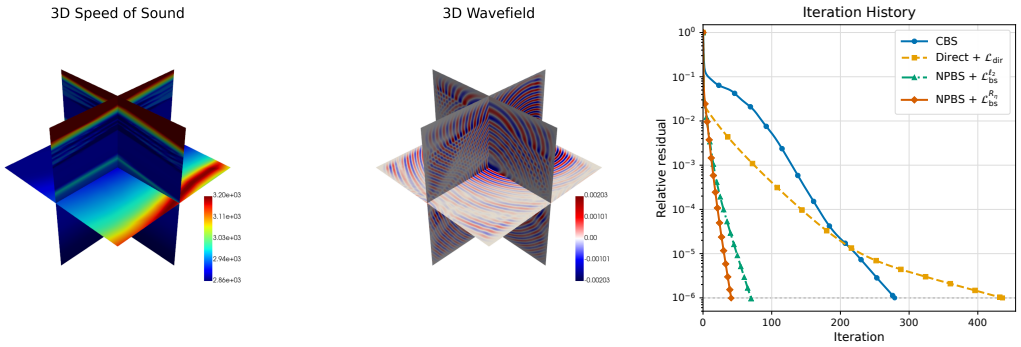

Figure 2: A three-dimensional Kimberlina-V1 test example. Left: slices of the velocity model. Middle: slices of the wavefield. Right: relative-residual histories for CBS and the three learned formulations.

5.2 Extensions Beyond the Helmholtz Equation

We apply the same iteration format and loss design to a linear convection–diffusion–reaction (CDR) system and Newton linear systems for a nonlinear PDE, using an appropriate reference operator and a correction network trained separately for each PDE setting. Dataset and setup details are in Appendix G.2 and F.

Linear Convection–Diffusion–Reaction (CDR) Systems We consider the linear CDR equation

$$-\nabla \cdot (\kappa \nabla u) + \mathbf{v} \cdot \nabla u + \sigma u = f \quad \text{in } \Omega \subset \mathbb{R}^d, \tag{22}$$

posed with periodic boundary conditions. Following the Helmholtz study, we compare three formulations and hold the architecture fixed so that only the iteration format and training objective vary. We construct a synthetic dataset on a 128×128 periodic grid with spatially heterogeneous coefficients (dataset generation details and representative samples are provided in Appendix F.2).

As shown in Table 5 and Figure 3, the advantage of NPBS with $\mathcal{L}_{\text{bs}}^{R_\eta}$ observed on the Helmholtz equation carries over to the CDR setting: the metric-matched formulation consistently achieves the fastest convergence among the three objectives.

Newton Linear Systems for Nonlinear PDEs We consider the non-convex problem with multiple solutions studied in [20]:

$$\begin{cases} -\Delta u(x,y) - u(x,y)^2 = -s \sin(\pi x) \sin(\pi y), & (x,y) \in \Omega, \\ u(x,y) = 0, & (x,y) \in \partial\Omega, \end{cases} \tag{23}$$

with $f(x,y) := -s \sin(\pi x) \sin(\pi y)$ and define nonlinear residual $\mathcal{F}_{\text{nl}}(u) := -\Delta u - u^2 - f$. At Newton iterate $u^{(m)}$, the linearized correction $\delta u^{(m)}$ solves

$$\begin{cases} -\Delta(\delta u^{(m)}) - 2u^{(m)}\, \delta u^{(m)} = -\mathcal{F}_{\text{nl}}(u^{(m)}), & (x,y) \in \Omega, \\ \delta u^{(m)} = 0, & (x,y) \in \partial\Omega, \end{cases} \tag{24}$$

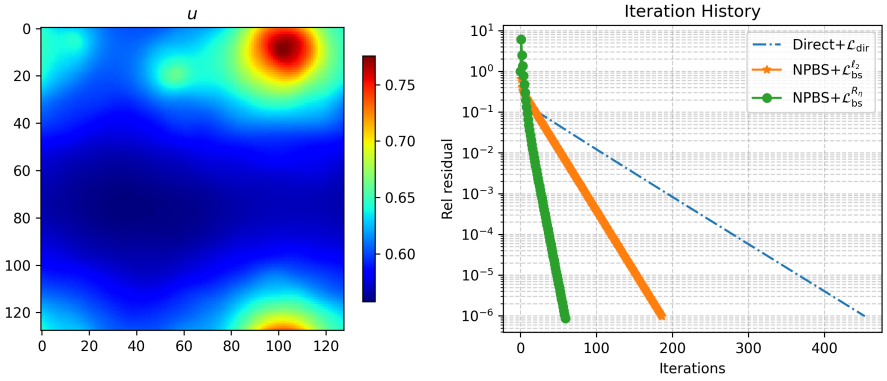

Figure 3: Representative CDR test case. Left: solution field u. Right: residual histories.

followed by the update $u^{(m+1)} = u^{(m)} + \delta u^{(m)}$. In every Newton step, the learned preconditioner is applied to the inner linear system (24). Following [20], discretization and dataset details are in Appendix F.3. For this Newton benchmark, the inner linear solves use $\mathrm{rTol} = 10^{-4}$.

Table 5 shows that the ranking observed in the linear benchmarks carries over to the Newton inner solves: the metric-matched NPBS requires the fewest iterations. Figure 4 further shows that the learned preconditioner integrates stably within the outer Newton loop.

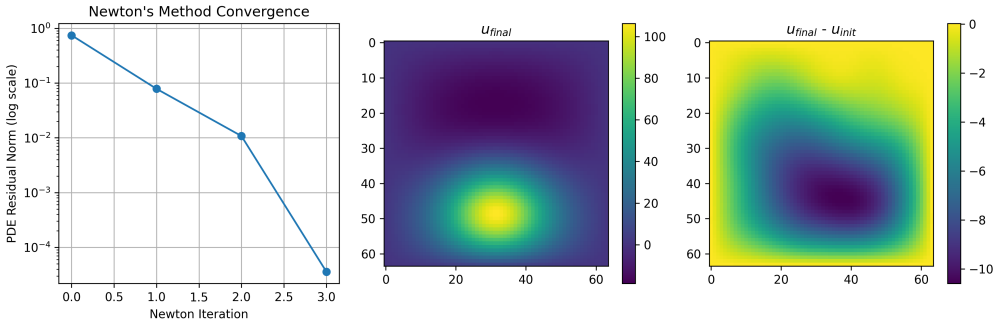

Figure 4: FNO preconditioner with NPBS+$\mathcal{L}_{\text{bs}}^{R_\eta}$ embedded in Newton iterations.

Table 5: Iteration format and loss comparison across other PDEs.

Method	**CDR Benchmark**		**Newton Inner-Solve**	
	Avg. Iters	Avg. Rel Res	Avg. Iters	Avg. Rel Res
Direct + $\mathcal{L}_{\text{dir}}$	391	9.84×10^{-7}	25	8.98×10^{-5}
NPBS + $\mathcal{L}_{\text{bs}}^{\ell_2}$	167	9.57×10^{-7}	11	7.20×10^{-5}
NPBS + $\mathcal{L}_{\text{bs}}^{R_\eta}$	**96**	8.68×10^{-7}	**9**	6.75×10^{-5}

6 Conclusion

We introduced Neural Preconditioned Born Series (NPBS), a learned iterative preconditioning framework that operates in Born-preconditioned residual coordinates. The identity $I - G_\eta V_\eta = G_\eta A$ shows that these coordinates are exactly those of shifted-Laplacian left preconditioning, and the induced metric $R_\eta = G_\eta^* G_\eta$ leads to a metric-matched objective $\mathcal{L}_{\text{bs}}^{R_\eta}$ that aligns training with inference. Experiments on heterogeneous Helmholtz problems show that metric-matched NPBS consistently reduces iteration counts, with larger gains in more difficult regimes and robustness across

neural backbones. FGMRES with NPBS gives the lowest solve time in the evaluated two-dimensional comparison. Results on CDR systems and Newton linear systems for nonlinear PDEs further indicate that residual-metric matching is useful beyond Helmholtz. Our current implementation is limited to structured Cartesian grids. Future work includes extending the reference inverse and correction networks to unstructured meshes, developing lighter neural preconditioners, and addressing time-dependent PDEs.

Acknowledgments and Disclosure of Funding

J.J. was supported by the focused project [FPF10120250014] of the Shenzhen Loop Area Institute and the National Natural Science Foundation of China [22341302]. X.L. was supported by the National Natural Science Foundation of China (Grant No. 12401571) and the Shandong Provincial Natural Science Foundation (Grant No. 2026HWYQ-024). The authors declare no competing interests.

References

[1] Yael Azulay and Eran Treister. Multigrid-augmented deep learning preconditioners for the helmholtz equation. *SIAM Journal on Scientific Computing*, 45(3):S127–S151, 2022.

[2] Ivo M Babuska and Stefan A Sauter. Is the pollution effect of the fem avoidable for the helmholtz equation considering high wave numbers? *SIAM Journal on numerical analysis*, 34 (6):2392–2423, 1997.

[3] Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 114(2):185–200, 1994.

[4] John P Boyd. *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[5] Achi Brandt and Irene Livshits. Wave-ray multigrid method for standing wave equations. *Electron. Trans. Numer. Anal*, 6(162-181):91, 1997.

[6] Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

[7] Shuhao Cao, Francesco Brarda, Ruipeng Li, and Yuanzhe Xi. Spectral-refiner: Accurate fine-tuning of spatiotemporal fourier neural operator for turbulent flows. In *International Conference on Learning Representations*, volume 2025, pages 60348–60384, 2025.

[8] Yuyan Chen, Bin Dong, and Jinchao Xu. Meta-mgnet: Meta multigrid networks for solving parameterized partial differential equations. *Journal of computational physics*, 455:110996, 2022.

[9] Zhiming Chen and Xueshuang Xiang. A source transfer domain decomposition method for helmholtz equations in unbounded domain. *SIAM Journal on Numerical Analysis*, 51(4): 2331–2356, 2013.

[10] Chen Cui, Kai Jiang, Yun Liu, and Shi Shu. A hybrid iterative neural solver based on spectral analysis for parametric pdes. *Journal of Computational Physics*, 538:114165, 2025.

[11] Chen Cui, Kai Jiang, and Shi Shu. A neural multigrid solver for helmholtz equations with high wavenumber and heterogeneous media. *SIAM Journal on Scientific Computing*, 47(3): C655–C679, 2025.

[12] Chengyuan Deng, Shihang Feng, Hanchen Wang, Xitong Zhang, Peng Jin, Yinan Feng, Qili Zeng, Yinpeng Chen, and Youzuo Lin. Openfwi: Large-scale multi-structural benchmark datasets for full waveform inversion. *Advances in Neural Information Processing Systems*, 35: 6007–6020, 2022.

[13] Nunzio Dimola, Alessandro Coclite, and Paolo Zunino. Neural preconditioning via krylov subspace geometry. *Bollettino dell'Unione Matematica Italiana*, pages 1–23, 2025.

[14] Yogi A Erlangga, Cornelis Vuik, and Cornelis Willebrordus Oosterlee. On a class of preconditioners for solving the helmholtz equation. *Applied Numerical Mathematics*, 50(3-4):409–425, 2004.

[15] Yogi A Erlangga, Cornelis W Oosterlee, and Cornelis Vuik. A novel multigrid based preconditioner for heterogeneous helmholtz problems. *SIAM Journal on Scientific Computing*, 27(4): 1471–1492, 2006.

[16] Oliver G Ernst and Martin J Gander. Why it is difficult to solve helmholtz problems with classical iterative methods. *Numerical analysis of multiscale problems*, pages 325–363, 2011.

[17] Martin J Gander and Hui Zhang. A class of iterative solvers for the helmholtz equation: Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized schwarz methods. *Siam Review*, 61(1):3–76, 2019.

[18] Ivan G Graham, Euan A Spence, and Jun Zou. Domain decomposition with local impedance conditions for the helmholtz equation with absorption. *SIAM Journal on Numerical Analysis*, 58(5):2515–2543, 2020.

[19] Daniel Greenfeld, Meirav Galun, Ronen Basri, Irad Yavneh, and Ron Kimmel. Learning to optimize multigrid pde solvers. In *International Conference on Machine Learning*, pages 2415–2423. PMLR, 2019.

[20] Wenrui Hao, Xinliang Liu, and Yahong Yang. Newton informed neural operator for solving nonlinear partial differential equations. *Advances in neural information processing systems*, 37: 120832–120860, 2024.

[21] Juncai He, Xinliang Liu, and Jinchao Xu. MgNO: Efficient parameterization of linear operators via multigrid. In *The Twelfth International Conference on Learning Representations*, 2024.

[22] Juncai He, Xinliang Liu, and Jinchao Xu. Self-composing neural operators for high-frequency and multiscale pde surrogates. *Journal of Computational Physics*, 565:115189, 2026. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2026.115189. URL `https://www.sciencedirect.com/science/article/pii/S0021999126005413`.

[23] Jun Hu and Pengzhan Jin. A hybrid iterative method based on mionet for pdes: Theory and numerical examples. *Mathematics of Computation*, 2025.

[24] Xingguo Huang and Stewart Greenhalgh. A finite-difference iterative solver of the helmholtz equation for frequency-domain seismic wave modeling and full-waveform inversion. *Geophysics*, 86(2):T107–T116, 2021.

[25] Moshe Israeli and Steven A Orszag. Approximation of radiation boundary conditions. *Journal of computational physics*, 41(1):115–135, 1981.

[26] Alena Kopaničáková, Youngkyu Lee, and George Em Karniadakis. Leveraging operator learning to accelerate convergence of the preconditioned conjugate gradient method. *Machine Learning for Computational Science and Engineering*, 1(2):39, 2025.

[27] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[28] Youngkyu Lee, Francesc Levrero Florencio, Jay Pathak, and George Em Karniadakis. Hybrid iterative solvers with geometry-aware neural preconditioners for parametric pdes. *arXiv preprint arXiv:2512.14596*, 2025.

[29] Youngkyu Lee, Shanqing Liu, Jerome Darbon, and George Em Karniadakis. A neural-operator preconditioned newton method for accelerated nonlinear solvers. *arXiv preprint arXiv:2511.08811*, 2025.

[30] Youngkyu Lee, Shanqing Liu, Zongren Zou, Adar Kahana, Eli Turkel, Rishikesh Ranade, Jay Pathak, and George Em Karniadakis. Fast meta-solvers for 3d complex-shape scatterers using neural operators trained on a non-scattering problem. *Computer Methods in Applied Mechanics and Engineering*, 446:118231, 2025.

[31] Bar Lerer, Ido Ben-Yair, and Eran Treister. Multigrid-augmented deep learning preconditioners for the helmholtz equation using compact implicit layers. *SIAM Journal on Scientific Computing*, 46(5):S123–S144, 2024.

[32] Yichen Li, Peter Yichen Chen, Tao Du, and Wojciech Matusik. Learning preconditioners for conjugate gradient pde solvers. In *International Conference on Machine Learning*, pages 19425–19439. PMLR, 2023.

[33] Zhihao Li, Di Xiao, Zhilu Lai, and Wei Wang. Neural preconditioning operator for efficient pde solves. *arXiv preprint arXiv:2502.01337*, 2025.

[34] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[35] Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023.

[36] Xinliang Liu, Bo Xu, Shuhao Cao, and Lei Zhang. Mitigating spectral bias for the multiscale operator learning. *Journal of Computational Physics*, 506:112944, 2024.

[37] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

[38] H.A. Melchers, J.H.M. Prins, and M.R.A. Abdelmalik. Neural green's operators for parametric partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 455: 118893, 2026. doi: https://doi.org/10.1016/j.cma.2026.118893.

[39] Gerwin Osnabrugge, Saroch Leedumrongwatthanakun, and Ivo M Vellekoop. A convergent born series for solving the inhomogeneous helmholtz equation in arbitrarily large media. *Journal of computational physics*, 322:113–124, 2016.

[40] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR, 2019.

[41] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[43] Alexander Rudikov, Vladimir Fanaskov, Ekaterina Muravleva, Yuri M. Laevsky, and Ivan Oseledets. Neural operators meet conjugate gradients: The FCG-NO method for efficient PDE solving. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 42766–42782. PMLR, 21–27 Jul 2024.

[44] Youcef Saad. A flexible inner-outer preconditioned gmres algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[45] Jie Shen, Tao Tang, and Li-Lian Wang. *Spectral methods: algorithms, analysis and applications*, volume 41. Springer Science & Business Media, 2011.

[46] Antonio Stanziola, Simon R Arridge, Ben T Cox, and Bradley E Treeby. A helmholtz equation solver using unsupervised learning: Application to transcranial ultrasound. *Journal of computational physics*, 441:110430, 2021.

[47] Antonio Stanziola, Simon Arridge, Ben T Cox, and Bradley E Treeby. A learned born series for highly-scattering media. *JASA Express Letters*, 3(5), 2023.

[48] Christiaan C Stolk. A dispersion minimizing scheme for the 3-d helmholtz equation based on ray theory. *Journal of computational Physics*, 314:618–646, 2016.

[49] Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361:112732, 2020.

[50] Qi Sun, Shengyan Li, Bowen Zheng, Lili Ju, and Xuejun Xu. Learning singularity-encoded green's functions with application to iterative methods. *Journal of Computational Physics*, 559: 114894, 2026. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2026.114894.

[51] Lloyd N Trefethen. *Spectral methods in MATLAB*. SIAM, 2000.

[52] Yan Xie, Minrui Lv, and Chen-Song Zhang. Mgcfnn: A neural multigrid solver with novel fourier neural network for high wave number helmholtz equations. In *The Thirteenth International Conference on Learning Representations*, 2025.

[53] Zherui Yang, Zhehao Li, Kangbo Lyu, Yixuan Li, Tao Du, and Ligang Liu. Learning sparse approximate inverse preconditioners for conjugate gradient solvers on GPUs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=jtMDzggo6M`.

[54] Zhijun Zeng, Youjia Zheng, Hao Hu, Zeyuan Dong, Yihang Zheng, Xinliang Liu, Jinzhuo Wang, Zuoqiang Shi, Linfeng Zhang, Yubing Li, et al. Openbreastus: Benchmarking neural operators for wave imaging using breast ultrasound computed tomography. *arXiv preprint arXiv:2507.15035*, 2025.

[55] Zhijun Zeng, Youjia Zheng, Chang Su, Qianhang Wu, Hao Hu, Zeyuan Dong, Yang Lv, Ligang Cui, Zhiyong Hou, Weijun Lin, Zuoqiang Shi, Yubing Li, and He Sun. Ultrasound tomography of musculoskeletal tissues with generative neural physics, 2026. URL `https://arxiv.org/abs/2508.12226`.

[56] Enrui Zhang, Adar Kahana, Alena Kopaničáková, Eli Turkel, Rishikesh Ranade, Jay Pathak, and George Em Karniadakis. Blending neural operators and relaxation methods in pde numerical solvers. *Nature Machine Intelligence*, 6(11):1303–1313, 2024.

A Proofs and Supporting Analysis

A.1 Proof of Proposition 4.1

Proof. Using (12), (19) can be rewritten as

$$\mathcal{L}_{\mathrm{bs}}^{R_\eta}(\theta) = \mathbb{E}_{\boldsymbol{r}\sim\mathcal{D}} \frac{\|G_\eta A\,\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - G_\eta \boldsymbol{r}\|_2}{\|G_\eta \boldsymbol{r}\|_2}. \tag{25}$$

By definition, $\|\boldsymbol{x}\|_{R_\eta} = \|G_\eta \boldsymbol{x}\|_2$. Substituting $\boldsymbol{x} = A\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - \boldsymbol{r}$ gives $\|A\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - \boldsymbol{r}\|_{R_\eta} = \|G_\eta A\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - G_\eta \boldsymbol{r}\|_2$, and similarly $\|\boldsymbol{r}\|_{R_\eta} = \|G_\eta \boldsymbol{r}\|_2$, yielding (21). □

A.2 Interpretation in preconditioned geometry.

Relative to (15), (18) and (19) replace the direct regression target A^{-1} with the preconditioned target $(G_\eta A)^{-1}$ by feeding $G_\eta \boldsymbol{r}$ into $\mathcal{M}_\theta$. The key difference lies in the residual metric: $\mathcal{L}_{\mathrm{bs}}^{\ell_2}$ measures the physical residual $e(\boldsymbol{r};\theta) := A\,\mathcal{M}_\theta(G_\eta \boldsymbol{r}) - \boldsymbol{r}$ in the Euclidean norm $\|e(\boldsymbol{r};\theta)\|_2$, whereas $\mathcal{L}_{\mathrm{bs}}^{R_\eta}$ measures it in the preconditioned norm $\|e(\boldsymbol{r};\theta)\|_{R_\eta} = \|G_\eta e(\boldsymbol{r};\theta)\|_2$. By Proposition 4.1, the latter is exactly the residual ratio evaluated in the same R_η-geometry induced by the preconditioned coordinates used by NPBS inference. For the constant-coefficient Helmholtz reference, the Fourier symbol

$$\widehat{G_\eta}(\boldsymbol{\xi}) = \frac{1}{|\boldsymbol{\xi}|^2 - k_0^2 - i\eta} \tag{26}$$

shows how the metric reweights residual modes through G_η.

A.3 Conditional convergence and bounded perturbations

Fix A and L_η, with L_η invertible and $G_\eta = L_\eta^{-1}$, and write $B_\eta = G_\eta A$ and $\boldsymbol{z}^m = G_\eta(\boldsymbol{f} - A\boldsymbol{u}^m)$. Equation (16) gives

$$\boldsymbol{z}^{m+1} = \boldsymbol{z}^m - B_\eta \mathcal{M}_\theta(\boldsymbol{z}^m). \tag{27}$$

If B_η is nonsingular and a uniform $0 \le \rho < 1$ satisfies

$$\|\boldsymbol{z} - B_\eta \mathcal{M}_\theta(\boldsymbol{z})\|_2 \le \rho \|\boldsymbol{z}\|_2 \tag{28}$$

at every residual encountered, then

$$\begin{aligned} \|\boldsymbol{z}^m\|_2 &\le \rho^m \|\boldsymbol{z}^0\|_2, \\ \|\boldsymbol{u}^m - \boldsymbol{u}^\star\|_2 &\le \|B_\eta^{-1}\|_2\, \rho^m \|\boldsymbol{z}^0\|_2. \end{aligned} \tag{29}$$

where $A\boldsymbol{u}^\star = \boldsymbol{f}$. Thus, uniform reduction of the Born-preconditioned residual implies geometric convergence to the solution. The loss (19) averages the corresponding one-step residual ratio.

Proof. Substitution of (16) into $\boldsymbol{z}^{m+1} = G_\eta(\boldsymbol{f} - A\boldsymbol{u}^{m+1})$ yields (27). Under (28), induction gives $\|\boldsymbol{z}^m\|_2 \le \rho^m \|\boldsymbol{z}^0\|_2$. If B_η is nonsingular, $\boldsymbol{z}^m = B_\eta(\boldsymbol{u}^\star - \boldsymbol{u}^m)$, which proves the solution-error bound in (29). Also, $\|\boldsymbol{f} - A\boldsymbol{u}^m\|_2 \le \|L_\eta\|_2 \rho^m \|\boldsymbol{z}^0\|_2$, so the physical residual converges for this fixed system. □

Bounded perturbations. For updates with additive perturbations,

$$\boldsymbol{z}^{m+1} = \boldsymbol{z}^m - B_\eta \mathcal{M}_\theta(\boldsymbol{z}^m) + \boldsymbol{d}^m, \qquad \|\boldsymbol{d}^m\|_2 \le \varepsilon, \tag{30}$$

assume that (28) holds at all encountered perturbed states with the same $\rho < 1$. The triangle inequality gives $\|\boldsymbol{z}^{m+1}\|_2 \le \rho \|\boldsymbol{z}^m\|_2 + \varepsilon$, and hence

$$\|\boldsymbol{z}^m\|_2 \le \rho^m \|\boldsymbol{z}^0\|_2 + \frac{1-\rho^m}{1-\rho}\varepsilon. \tag{31}$$

This is a conditional bound on accumulated residual perturbations. Minimizing the expected one-step residual ratio in (19) does not establish the uniform condition (28) on inference trajectories or guarantee stability on unseen problems. The finite-test contraction statistics in Appendix J are empirical evidence rather than a global guarantee.

A.4 Conditional spectral bounds

Under the assumptions of Proposition 3.1, if additionally $\|G_\eta V_\eta\|_2 \le q < 1$, then

$$\sigma(I - G_\eta V_\eta) \subset \{z \in \mathbb{C} : |z - 1| \le q\}, \tag{32}$$

and

$$\kappa_2(I - G_\eta V_\eta) \le \frac{1+q}{1-q}. \tag{33}$$

Proof. Since $\rho(G_\eta V_\eta) \le \|G_\eta V_\eta\|_2 \le q$, every eigenvalue μ of $G_\eta V_\eta$ satisfies $|\mu| \le q$. Eigenvalues of $I - G_\eta V_\eta$ are $1 - \mu$, giving (32). The condition $\|G_\eta V_\eta\|_2 < 1$ ensures the Neumann series converges:

$$(I - G_\eta V_\eta)^{-1} = \sum_{j=0}^{\infty} (G_\eta V_\eta)^j, \tag{34}$$

so that

$$\|I - G_\eta V_\eta\|_2 \le 1 + q, \qquad \|(I - G_\eta V_\eta)^{-1}\|_2 \le \sum_{j \ge 0} q^j = \frac{1}{1-q},$$

which yields (33). □

See also [24] for related analysis. The identity (12) and the metric $R_\eta = G_\eta^* G_\eta$ require only an invertible L_η; the bounds (32)–(33) require the additional norm condition. The norm estimates in Appendix J.1 do not support invoking these bounds to explain the observed performance. Exceeding one removes this spectral guarantee; it does not by itself imply divergence of the learned iteration.

A.5 Role of η in Metric and Iteration Coordinates

The shift η defines both the left preconditioner $G_\eta := L_\eta^{-1}$ and the induced residual metric $R_\eta := G_\eta^* G_\eta$. Consequently, the R_η-weighted residual norm satisfies

$$\|\boldsymbol{f} - A\boldsymbol{u}\|_{R_\eta}^2 := (\boldsymbol{f} - A\boldsymbol{u})^* R_\eta (\boldsymbol{f} - A\boldsymbol{u}) = \|G_\eta(\boldsymbol{f} - A\boldsymbol{u})\|_2^2, \tag{35}$$

so η controls the geometry in which residuals are measured and corrected, not only numerical damping. In practice, we tune η via a small validation sweep and then keep it fixed between training and inference for each dataset to preserve objective–inference consistency. Heuristically, increasing η can move the reference inverse farther from resonance and may improve the contraction behavior of the fixed-point map, whereas excessively large damping can also weaken the usefulness of the resulting correction directions. The sensitivity experiment in Appendix J.2 quantifies this trade-off around the canonical CBS scale.

B Full Derivations for CDR and Nonlinear Extensions

This appendix gives the full technical derivations deferred from the main text.

B.1 Convection-Diffusion-Reaction (Linear)

Consider

$$\mathcal{L}_{\mathrm{cdr}} u = -\nabla \cdot (\kappa(x) \nabla u) \; + \; \mathbf{v}(x) \cdot \nabla u \; + \; \sigma(x) u = f. \tag{36}$$

Choose background constants $\kappa_0, \mathbf{v}_0, \sigma_0$ and define

$$\kappa = \kappa_0 + \delta\kappa, \qquad \mathbf{v} = \mathbf{v}_0 + \delta\mathbf{v}, \qquad \sigma = \sigma_0 + \delta\sigma. \tag{37}$$

Set

$$\mathcal{L}_{0,\mathrm{cdr}} u := -\kappa_0 \Delta u \; + \; \mathbf{v}_0 \cdot \nabla u \; + \; \sigma_0 u, \tag{38}$$

and define $\mathcal{V}_{\mathrm{cdr}}$ by $\mathcal{L}_{\mathrm{cdr}} = \mathcal{L}_{0,\mathrm{cdr}} - \mathcal{V}_{\mathrm{cdr}}$, i.e.,

$$\mathcal{V}_{\mathrm{cdr}} u = \nabla \cdot (\delta\kappa \nabla u) \; - \; \delta\mathbf{v} \cdot \nabla u \; - \; \delta\sigma \, u. \tag{39}$$

Then

$$\mathcal{L}_{0,\mathrm{cdr}} u = \mathcal{V}_{\mathrm{cdr}} u + f. \tag{40}$$

With $G_{\mathrm{cdr}} := \mathcal{L}_{0,\mathrm{cdr}}^{-1}$:

$$(I - G_{\mathrm{cdr}} \mathcal{V}_{\mathrm{cdr}}) u = G_{\mathrm{cdr}} f = G_{\mathrm{cdr}} \mathcal{L}_{\mathrm{cdr}} u. \tag{41}$$

In discrete form $(A_{\mathrm{cdr}}, V_{\mathrm{cdr}}, G_{\mathrm{cdr}})$:

$$I - G_{\mathrm{cdr}} V_{\mathrm{cdr}} = G_{\mathrm{cdr}} A_{\mathrm{cdr}}. \tag{42}$$

Therefore, the CDR-specific instance of the metric-matched Born Series inspired objective is

$$\mathcal{L}_{\mathrm{bs}}^{R_{\mathrm{cdr}}}(\theta) = \mathbb{E}_{r \sim \mathcal{D}} \frac{\| (I - G_{\mathrm{cdr}} V_{\mathrm{cdr}}) \mathcal{M}_\theta(G_{\mathrm{cdr}} r) - G_{\mathrm{cdr}} r \|_2}{\| G_{\mathrm{cdr}} r \|_2} \tag{43}$$

or equivalently

$$\mathcal{L}_{\mathrm{bs}}^{R_{\mathrm{cdr}}}(\theta) = \mathbb{E}_{r \sim \mathcal{D}} \frac{\| G_{\mathrm{cdr}} A_{\mathrm{cdr}} \mathcal{M}_\theta(G_{\mathrm{cdr}} r) - G_{\mathrm{cdr}} r \|_2}{\| G_{\mathrm{cdr}} r \|_2}. \tag{44}$$

with $R_{\mathrm{cdr}} := G_{\mathrm{cdr}}^* G_{\mathrm{cdr}}$. This is exactly the same template as $\mathcal{L}_{\mathrm{bs}}^{R_\eta}$ in Section 4.2, under the substitution $(G_\eta, A, V_\eta) \mapsto (G_{\mathrm{cdr}}, A_{\mathrm{cdr}}, V_{\mathrm{cdr}})$.

For periodic domains, G_{cdr} is a convolution operator with Fourier symbol

$$\widehat{G}_{\mathrm{cdr}}(\xi) = \frac{1}{\kappa_0 |\xi|^2 + i \mathbf{v}_0 \cdot \xi + \sigma_0}, \tag{45}$$

so high-frequency diffusion-dominated modes are attenuated in training. As in Helmholtz, the objective measures the residual in the geometry of the preconditioned system.

B.2 Nonlinear PDE via Newton-Born Linearization

For the nonlinear example

$$\mathcal{F}_{\mathrm{nl}}(u) := -\Delta u - u^2 - f = 0, \tag{46}$$

Newton at iterate $u^{(m)}$ solves

$$J_m \, \delta u^{(m)} = r^{(m)}, \qquad J_m := -\Delta - 2u^{(m)}, \qquad r^{(m)} := -\mathcal{F}_{\mathrm{nl}}(u^{(m)}), \tag{47}$$

followed by $u^{(m+1)} = u^{(m)} + \delta u^{(m)}$.

Discrete realization used in this work (Dirichlet + FDM + DST). For the Newton experiments in Section 5, we use homogeneous Dirichlet boundary conditions on $\Omega = [0,1]^2$ and a five-point finite-difference discretization on interior unknowns. With $h = 1/(N+1)$, define the discrete negative Laplacian

$$L_{\mathrm{D}} u = \frac{1}{h^2} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} * u, \tag{48}$$

so the discrete Jacobian is

$$J_m^{\mathrm{disc}} = L_{\mathrm{D}} - 2 \, \mathrm{diag}(u^{(m)}). \tag{49}$$

In the following discrete formulation, we write $J_m = J_m^{\mathrm{disc}}$ and use the discrete Dirichlet Poisson operator L_{D} as the fixed reference, with inverse

$$G := L_{\mathrm{D}}^{-1}. \tag{50}$$

The Jacobian admits the split $J_m = L_{\mathrm{D}} - V_m$, where

$$V_m := L_{\mathrm{D}} - J_m = 2 \, \mathrm{diag}(u^{(m)}). \tag{51}$$

The fixed reference inverse G is applied using the DST-I. The identity $I - GV_m = GJ_m$ gives the preconditioned correction equation

$$(I - GV_m) \, \delta u^{(m)} = G r^{(m)}. \tag{52}$$

Hence the Newton-state instance of the metric-matched Born Series objective is

$$\mathcal{L}_{\mathrm{bs}}^{R}(\theta) = \mathbb{E}_{(u^{(m)},r)\sim\mathcal{D}_{\mathrm{N}}} \frac{\|(I - GV_m)\mathcal{M}_\theta(Gr) - Gr\|_2}{\|Gr\|_2}, \tag{53}$$

equivalently

$$\mathcal{L}_{\mathrm{bs}}^{R}(\theta) = \mathbb{E}_{(u^{(m)},r)\sim\mathcal{D}_{\mathrm{N}}} \frac{\|GJ_m\mathcal{M}_\theta(Gr) - Gr\|_2}{\|Gr\|_2}. \tag{54}$$

Here $R := G^*G$ is fixed across Newton steps, while J_m and the state-conditioned correction map $\mathcal{M}_\theta$ depend on $u^{(m)}$.

Discussion (nonlinear). The loss is computed on *linearized* systems along Newton trajectories, not directly on the nonlinear map. This has two effects:

1. It preserves the same preconditioned formulation as in the linear case.
2. It aligns training with actual inference, where the solver repeatedly tackles Jacobian systems with changing $u^{(m)}$.

In practice, one may use an inner iteration

$$\delta u_{k+1}^{(m)} = \delta u_k^{(m)} \; + \; \mathcal{M}_\theta\Big(G\left(r^{(m)} - J_m\delta u_k^{(m)}\right)\Big), \tag{55}$$

then set $\delta u^{(m)} \approx \delta u_K^{(m)}$ and update $u^{(m+1)} = u^{(m)} + \delta u^{(m)}$.

C Verification: Fast Transform Implementation of Green Operators

We verify, case by case, when Green-operator application is fast. A concrete Fourier pseudospectral discretization recipe for L_η and G_η is provided in Appendix D.

Proposition C.1 (Transform-diagonalizable Green operators)**.** *On a uniform grid, if the chosen reference operator has constant coefficients and compatible boundary conditions (periodic, or transform-compatible Dirichlet/Neumann), then its discrete matrix is diagonalizable by a fast trigonometric transform. Consequently, applying the corresponding discrete Green operator costs $\mathcal{O}(N \log N)$.*

Constructive verification. Let L_0 be any constant-coefficient reference operator and $G_0 = L_0^{-1}$. Under periodic boundary conditions, the discrete L_0 is block-circulant with circulant blocks, hence

$$L_0 = F^{-1}\Lambda F, \qquad G_0 = F^{-1}\Lambda^{-1}F,$$

where F is the multidimensional DFT matrix and Λ contains the symbol values. Therefore

$$G_0 q = \mathcal{F}^{-1}\left(\frac{\widehat{q}}{\lambda(\xi)}\right),$$

implemented by FFT/IFFT in $\mathcal{O}(N \log N)$. For Dirichlet or Neumann boundaries, the same statement holds with sine/cosine transforms (DST/DCT), which are FFT-based and have the same complexity order. □

Helmholtz case. With $L_\eta = -\Delta - (k_0^2 + i\eta)$, the symbol is

$$\lambda_{\mathrm{H}}(\xi) = |\xi|^2 - (k_0^2 + i\eta),$$

so

$$G_\eta q = \mathcal{F}^{-1}\left(\frac{\widehat{q}}{\lambda_{\mathrm{H}}(\xi)}\right).$$

Hence the Born residual $G_\eta r$ is FFT-fast under the stated boundary setting.

CDR case. With $\mathcal{L}_{0,\mathrm{cdr}} = -\kappa_0\Delta + \mathbf{v}_0 \cdot \nabla + \sigma_0$, the symbol is exactly (45) denominator:

$$\lambda_{\mathrm{cdr}}(\xi) = \kappa_0|\xi|^2 + i\mathbf{v}_0 \cdot \xi + \sigma_0.$$

Thus

$$G_{\mathrm{cdr}}q = \mathcal{F}^{-1}\left(\frac{\widehat{q}}{\lambda_{\mathrm{cdr}}(\xi)}\right),$$

again $\mathcal{O}(N \log N)$.

Newton linear systems for nonlinear PDEs. For the Newton experiments, we use homogeneous Dirichlet boundaries and five-point FDM on interior nodes. The discrete Dirichlet Poisson reference L_{D} is diagonalized by the 2D DST-I. Writing

$$L_{\mathrm{D}} = S^{-1}\Lambda_{\mathrm{N}}S,$$

with S the 2D DST-I transform, its inverse acts as

$$Gq = S^{-1}\left(\frac{Sq}{\lambda_{p,q}}\right),$$

where the eigenvalues of L_{D} are

$$\lambda_{p,q} = \frac{4}{h^2}\left[\sin^2\left(\frac{p\pi h}{2}\right) + \sin^2\left(\frac{q\pi h}{2}\right)\right].$$

The Newton reference inverse is therefore applied by DST/IDST in $\mathcal{O}(N \log N)$.

Boundary-condition note for all cases. In scattering settings, we use an absorbing layer (e.g., sponge/PML-style damping) on a finite box and choose a *constant-coefficient* reference operator on that computational domain; this is the operator inverted by FFT in the Born-inspired loss/iteration. If strict Dirichlet/Neumann boundaries are required, use DST/DCT (or FFT-based embedding), preserving near-identical complexity. In particular, the experiments on Newton linear systems for nonlinear PDEs use homogeneous Dirichlet boundaries and DST-based inversion of the discrete Dirichlet Poisson reference L_{D}.

D Fourier Pseudospectral Discretization of L_η and G_η

This appendix gives the concrete discretization used when L_η and $G_\eta = L_\eta^{-1}$ are implemented by FFT under periodic boundary conditions. In numerical-analysis terminology, this is a *Fourier pseudospectral* (or Fourier collocation) discretization: derivatives are represented by exact Fourier multipliers, while coefficient multiplications are applied pointwise in physical space [4, 51, 45].

Periodic grid and Fourier modes. Let $\Omega = \prod_{\alpha=1}^{d}[0, L_\alpha)$ and use a uniform grid with N_α points in direction α:

$$x_{j_\alpha} = j_\alpha h_\alpha, \qquad h_\alpha = L_\alpha/N_\alpha, \qquad j_\alpha = 0, \ldots, N_\alpha - 1.$$

For grid field $u_j = u(x_j)$, define the DFT coefficients $\widehat{u}_n$ with multi-index

$$n = (n_1, \ldots, n_d), \qquad n_\alpha \in \Big\{-\frac{N_\alpha}{2}, \ldots, \frac{N_\alpha}{2} - 1\Big\},$$

and physical wavenumbers

$$\xi_{\alpha,n} = \frac{2\pi n_\alpha}{L_\alpha}, \qquad |\xi_n|^2 = \sum_{\alpha=1}^{d}\xi_{\alpha,n}^2.$$

Discrete symbol of L_η. For

$$L_\eta = -\Delta - (k_0^2 + i\eta), \qquad \eta > 0,$$

the Fourier symbol is

$$\lambda_n = |\xi_n|^2 - (k_0^2 + i\eta),$$

so

$$\widehat{L_\eta u}_n = \lambda_n\,\widehat{u}_n.$$

Hence L_η is diagonal in Fourier space, and its inverse acts modewise:

$$\widehat{(G_\eta q)}_n = \frac{\widehat{q}_n}{\lambda_n}.$$

Because $\eta > 0$, λ_n has nonzero imaginary part and the modal inversion is well-defined.

FFT implementation of $G_\eta q$. Given q, compute:

1. $\widehat{q} = \mathcal{F}(q)$ (FFT),
2. $\widehat{z}_n = \widehat{q}_n / \lambda_n$ (pointwise complex division),
3. $z = \mathcal{F}^{-1}(\widehat{z})$ (IFFT).

Then $z = G_\eta q$. The complexity is $\mathcal{O}(N \log N)$ with $N = \prod_\alpha N_\alpha$.

Relation to A and V_η. For heterogeneous Helmholtz,

$$Au = -\Delta u - k(x)^2 u, \qquad V_\eta u = (k(x)^2 - k_0^2 - i\eta)u.$$

In Fourier pseudospectral form:

$$-\Delta u = \mathcal{F}^{-1}\big(|\xi|^2 \widehat{u}\big),$$

while $k(x)^2 u$ and $V_\eta u$ are computed pointwise in physical space. This derivative-in-Fourier / product-in-physical split is exactly the pseudospectral pattern [4, 51]. For strongly non-smooth coefficients, standard de-aliasing (e.g., $2/3$ truncation) can be used [45].

Boundary-condition scope. The FFT diagonalization above is exact for periodic boundaries. For homogeneous Dirichlet/Neumann boundaries, the same constant-coefficient reference operators are diagonalized by sine/cosine transforms (DST/DCT), with the same $\mathcal{O}(N \log N)$ complexity class. In this manuscript, G_η in CBS and NPBS is implemented with the periodic/transform-compatible constant-coefficient reference operator to preserve fast application.

E Architectural Instantiations

NPBS is not architecture-centered; it is compatible with different neural operators. For reproducibility, we briefly record the primary FNO instantiation used in most experiments.

FNO instantiation and design principle. Across Helmholtz, CDR, and Newton linear systems for nonlinear PDEs, we use an FNO-based preconditioner $\mathcal{M}_\theta$. Let a denote PDE coefficient fields (e.g., v in Helmholtz, or $a = (\kappa, \mathbf{v}, \sigma)$ in CDR), and let r denote the iterative residual. The design explicitly applies a *nonlinear* transform to coefficient features and a *linear* transform to residual features, then fuses both branches:

$$\mathcal{M}_\theta(r; a) = \mathcal{P}\Big(\mathrm{LFL}\Big[\mathrm{NLFL}(\Phi_\theta(a)) \odot \mathrm{LFL}(\Psi_\theta(r))\Big]\Big). \tag{56}$$

The lifting/project mappings are

$$\Phi_\theta(a) = W_a * \mathrm{LayerNorm}(\mathrm{pad}(a)), \qquad \Psi_\theta(r) = W_r * \mathrm{pad}(r), \tag{57}$$

and

$$\mathcal{P}(z) = W_p * \mathrm{unpad}(z). \tag{58}$$

Here $\mathrm{NLFL}(\Phi_\theta(a))$ is used for nonlinear coefficient encoding, while the residual branch keeps linear processing through $\mathrm{LFL}(\Psi_\theta(r))$, ensuring linearity with respect to r.

The Fourier blocks are

$$\mathrm{LFL}(x) = \mathcal{F}^{-1}\big(\mathcal{W}_\theta^{\mathrm{F}} \odot \mathcal{F}(x)\big) + \mathcal{W}_\theta^{\mathrm{loc}} * x, \tag{59}$$

$$\mathrm{NLFL}(x) = \varphi\big(\mathrm{LFL}(x)\big), \tag{60}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ are Fourier transforms, $\mathcal{W}_\theta^{\mathrm{F}}$ are truncated-mode spectral weights, $\mathcal{W}_\theta^{\mathrm{loc}}$ is a local 1×1 convolution, and φ is GELU. In implementation, coefficient features pass through stacked NLFL blocks, while the residual branch uses a single LFL block.

F Dataset Illustration

F.1 Heterogeneous Helmholtz Equation

We use velocity models from the OpenFWI benchmark [12]. For both CurveVel-A and CurveFault-B, the training set contains $3{,}000$ samples and the test set contains 50 samples. The computational domain is discretized on a 256×256 grid with a 32-cell perfectly matched layer (PML) on each side, yielding a padded grid of size 320×320. Spatial velocity fields range from approximately 1500 to 4500 m/s, and the grid resolution is approximately 12 points per wavelength. Representative velocity models are shown in Figure 5.

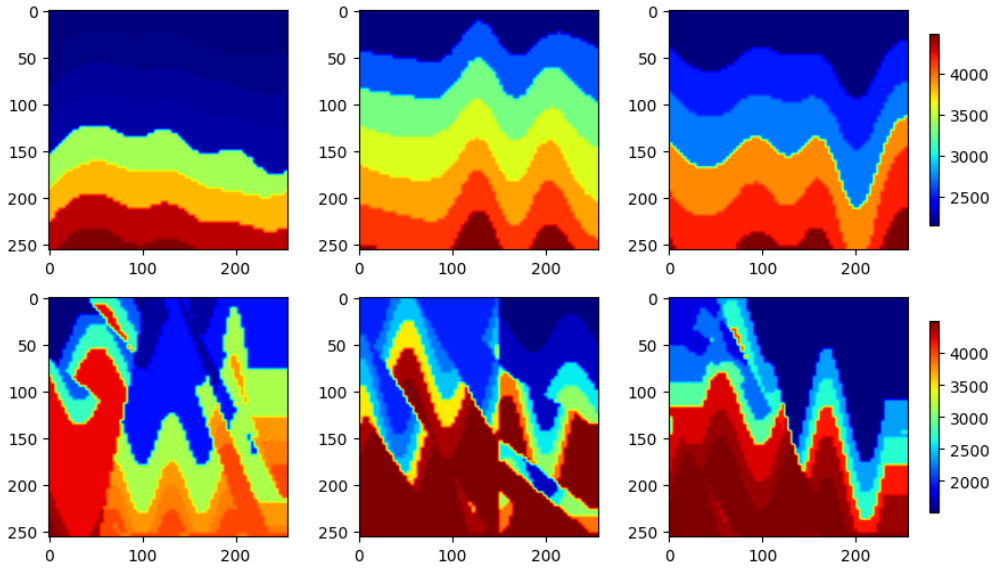

Figure 5: Representative heterogeneous velocity models from the OpenFWI datasets. Top row: CurveVel-A, featuring relatively smooth, curved velocity transitions. Bottom row: CurveFault-B, characterized by sharp discontinuities, strong contrasts, and complex fault structures.

F.2 CDR Dataset Generation

The CDR benchmark uses a 128×128 periodic grid. The training set contains 4000 samples and the test set contains 400 samples. The diffusion field κ is drawn from a Gaussian random field (GRF); the velocity $\mathbf{v}$ is obtained from a GRF stream function ψ via

$$v_x = \frac{\partial \psi}{\partial y}, \qquad v_y = -\frac{\partial \psi}{\partial x}, \tag{61}$$

so that incompressibility $\nabla \cdot \mathbf{v} = 0$ is enforced by construction; the source term f is a Gaussian bump with randomly sampled center and width. Representative samples are shown in Figure 6.

F.3 Newton Benchmark Data

Following [20], the domain $\Omega = [0, 1]^2$ is discretized with $N = 64$ interior grid points (grid spacing $h = 1/65$) and source amplitude $s = 1600$. Training data consists of $10{,}000$ Newton trajectory pairs (combining two trajectory sets of $5{,}000$ each from different solution branches), and the test set contains $1{,}000$ samples. Supervision is obtained by generating Newton trajectories with sparse direct solves. To ensure that the trajectories probe distinct solution branches, initial guesses are constructed by perturbing a reference solution with low-frequency sine-series noise,

$$u^0 = u^\star + \alpha \sum_{i,j=1}^{4} c_{ij} \sin(i\pi x) \sin(j\pi y), \quad c_{ij} \sim \mathcal{N}(0, 0.1^2), \ \alpha = 4.0. \tag{62}$$

Representative branches obtained in this way are shown in Figure 7.

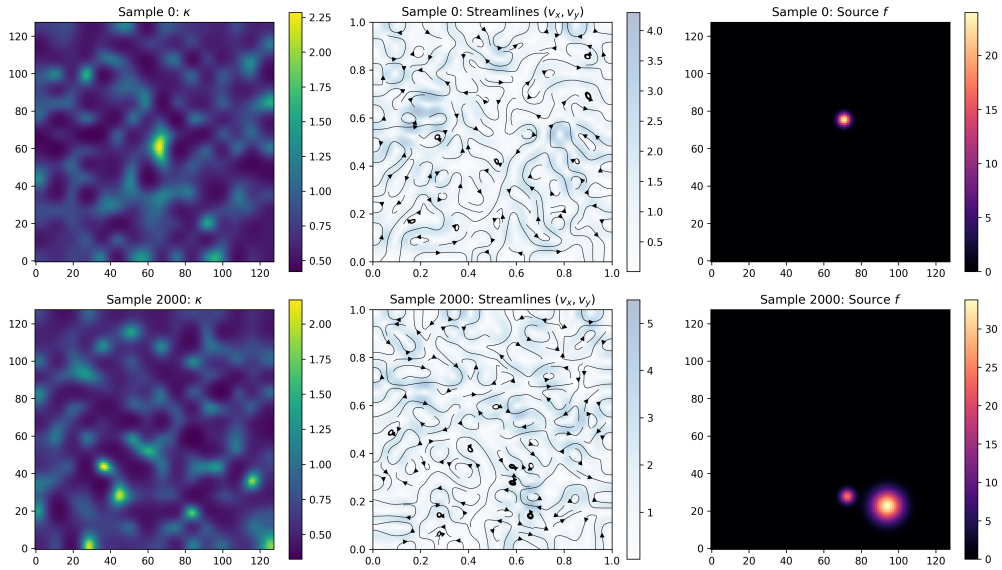

Figure 6: Representative samples from the CDR benchmark. Each row shows the diffusion field κ, the incompressible velocity field $\mathbf{v}$ visualized by streamlines, and the source term f.

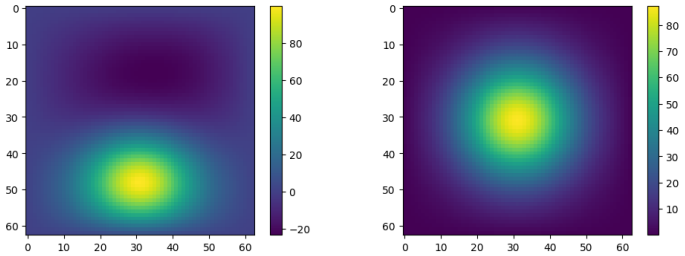

Figure 7: Multiple solutions of the nonlinear PDE under different initial guesses.

G Experimental Setup

This appendix details the dataset splits, backbone architectures, and training hyperparameters for all experiments reported in Section 5.

G.1 Heterogeneous Helmholtz Equation

Dataset. For the two-dimensional experiments, we use velocity models from the OpenFWI benchmark [12]. For both CurveVel-A and CurveFault-B, the training set contains $3{,}000$ samples and the test set contains 50 samples. The computational domain is discretized on a 256×256 grid with a 32-cell perfectly matched layer (PML) on each side, yielding a padded grid of size 320×320.

FNO backbone (primary). In the two-dimensional study, the primary preconditioner follows the FNO instantiation of Eq. (56) with 3 NLFL layers, 25 hidden channels, and 45 retained Fourier modes per layer. The residual branch uses a single LFL with full-bandwidth modes ($N_{\mathrm{pad}}/2 + 1$). Training uses the AdamW optimizer for 400 epochs with batch size 30, initial learning rate 10^{-3}, and a step-decay schedule (factor 0.8 every 20 epochs). Inference is performed in float64 with a relative tolerance of 10^{-6} and a maximum of $2{,}500$ iterations.

Comparison backbones. To verify that the NPBS gains are architecture-agnostic, we evaluate additional backbones under the same NPBS+$\mathcal{L}_{\mathrm{bs}}^{R_\eta}$ framework. In the two-dimensional comparisons, MgNO, U-Net, Encoder–solver, and MGCFNN are trained with AdamW for 200 epochs, with batch

size 30, initial learning rate 10^{-3}, and a step-decay schedule (factor 0.8 every 10 epochs). Their architecture-specific parameters are:

- **MgNO** [21]: A multigrid neural operator with 32 channels, 4-level V-cycle structure with $[2, 2]$ pre-/post-smoothing iterations per level, and GELU activation.
- **MGCFNN** [52]: A multigrid convolutional Fourier neural network with 32 channels, 3-level hierarchy. Level 0 uses spatial 5×5 convolutions (1 sweep); levels 1 and 2 use complex Fourier layers with wavenumber cutoffs 15 and 10 respectively (2 and 4 sweeps).
- **Encoder-Solver** [31]: A U-Net-style encoder-decoder with 32 channels, 4 levels, and implicit weight-sharing across decoder iterations.
- **U-Net** [42]: A standard 4-level U-Net with 32 channels and skip connections; the rhs-path convolutions are bias-free to preserve linearity in the residual.

For runtime, we use `torch.compile` when stable. MGCFNN is evaluated without compilation because `torch.compile` led to divergence during inference. The original Encoder–solver diverged in our iterative tests; we therefore use a variant that is linear in the residual and retains its core multigrid-inspired components.

Three-dimensional Helmholtz setup. The OpenFWI Kimberlina-V1 velocity volumes have native resolution $350 \times 400 \times 400$ with $10\,\mathrm{m}$ grid spacing and a physical extent of approximately $3.5 \times 4 \times 4\,\mathrm{km}$. We interpolate them to $192 \times 192 \times 192$ grids. The training set contains 600 samples from the first 35 years, while the test set contains 50 samples from year 100. We use MgNO with three-dimensional convolutions and FFTs, an absorbing-boundary width of 16 cells, PPW 6, and 200 training epochs per formulation.

G.2 Extensions Beyond the Helmholtz Equation

G.2.1 Convection–Diffusion–Reaction (CDR)

Dataset. The CDR benchmark uses a 128×128 periodic grid. The training set contains $4{,}000$ samples and the test set contains 400 samples. Coefficients are generated as described in Appendix F.2.

FNO backbone. The preconditioner uses the same FNO design principle as in the Helmholtz setting (Eq. (56)), with 25 hidden channels, 3 NLFL layers with 32 retained Fourier modes per layer, and zero-padding of 8 cells. Training uses Adam for 60 epochs with batch size 32, initial learning rate 10^{-3}, and step-decay (factor 0.8 every 3 epochs). Inference uses a relative tolerance of 10^{-6}.

G.2.2 Newton Linear Systems for Nonlinear PDEs

Dataset. Following [20], the domain $\Omega = [0, 1]^2$ is discretized with $N = 64$ interior grid points (grid spacing $h = 1/65$) and the source amplitude is $s = 1600$. Training data consists of $10{,}000$ Newton trajectory pairs (combining two trajectory sets of $5{,}000$ each from different solution branches), and the test set contains $1{,}000$ samples.

FNO backbone. The preconditioner uses 25 hidden channels, 2 NLFL layers with 20 retained Fourier modes, and zero-padding of 8 cells. Training uses Adam for 200 epochs with batch size 50, initial learning rate 10^{-4}, and weight decay 10^{-6}. The inner linear solver uses a relative tolerance of 10^{-4}.

H Compute Resources

All experiments were conducted using PyTorch 2.6.0 with CUDA 12.4 on a single workstation equipped with an Intel Xeon Platinum 8468 CPU and one NVIDIA H100 GPU. All solvers—including classical baselines such as CBS and the shifted-Laplacian preconditioning—are implemented entirely in PyTorch and accelerated via CUDA, ensuring that wall-clock comparisons reflect algorithmic differences rather than implementation-level disparities. No distributed training, large-scale pretraining, or fine-tuning of pretrained foundation models was involved; the main computational cost comes from training lightweight neural preconditioners and iterative solver inference.

I Training Convergence

Figure 8 shows the training and validation rTol histories for the three loss configurations on CurveFault-B.

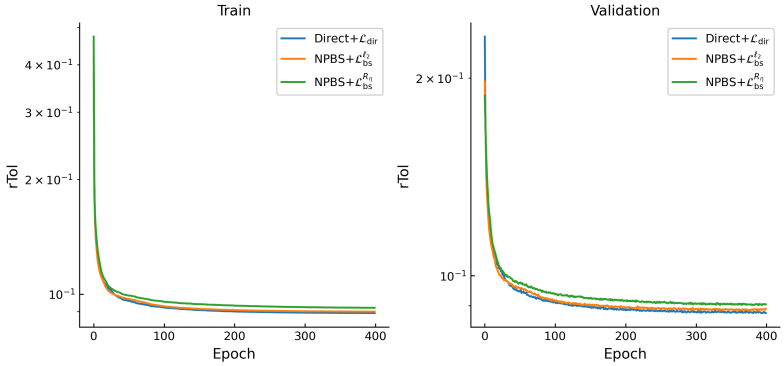

Figure 8: Training and validation rTol histories on CurveFault-B.

Table 6: Training statistics on CurveFault-B under the same FNO setup.

Method	Train rTol	Val rTol	Train time
Direct + $\mathcal{L}_{\text{dir}}$	8.91×10^{-2}	8.77×10^{-2}	52 min 18 s
NPBS + $\mathcal{L}_{\text{bs}}^{\ell_2}$	8.99×10^{-2}	8.90×10^{-2}	52 min 35 s
NPBS + $\mathcal{L}_{\text{bs}}^{R_\eta}$	9.20×10^{-2}	9.04×10^{-2}	52 min 39 s

J Additional Helmholtz diagnostics and evaluation

J.1 Checking the spectral assumption

Table 7 reports the estimated operator norms for the two OpenFWI datasets. The diagnostic uses 1,000 randomly selected samples and reports the mean estimate for each dataset and PPW. All reported means exceed one, so the sufficient condition for the spectral bounds in Appendix A.4 cannot be invoked as a guarantee for these experiments. These are empirical norm estimates, not certified spectral bounds; the reported means do not assert that every sample has the same behavior. The algebraic identity and the induced training metric remain valid.

Table 7: Estimated $\|G_\eta V_\eta\|_2$ (mean).

Dataset	PPW	Frequency (Hz)	$\|G_\eta V_\eta\|_2$
CurveFault-B	12	92.0	1.3705
	14	78.9	1.3729
	16	69.0	1.3747
	24	46.0	1.3785
CurveVel-A	12	92.0	1.2881
	14	78.9	1.2952
	16	69.0	1.3020
	24	46.0	1.3246

J.2 Sensitivity to η

We use the canonical CBS scale $\eta_{\text{base}} = \max_x |k(x)^2 - k_0^2|$ as a reference and vary the ratio η/η_{base}. Table 8 reports the average NPBS iteration counts for the tested values. The iteration diverges at

$0.1\eta_{\text{base}}$, whereas all reported scales from $0.2\eta_{\text{base}}$ to $100\eta_{\text{base}}$ converge. The fewest iterations occur at $\eta = \eta_{\text{base}}$; increasing the shift beyond this value progressively raises the iteration count. These results support choosing η near the canonical scale for this experiment.

Table 8: Sensitivity of NPBS to the shift η. The symbol $\times$ denotes divergence at the tested value.

η/η_{base}	0.1	0.2	0.4	0.8	1	1.4	2	5	10	100
Avg. Iters	$\times$	545	192	127	**126**	137	151	166	196	283

J.3 Empirical contraction diagnostics

The maximum contraction factors recorded for the MgNO tests are summarized in Table 9. Metric-matched NPBS has the smallest mean among the three formulations. These finite-test statistics do not establish a uniform contraction constant for all residuals or all test trajectories, and are distinct from the sufficient condition in (28).

Table 9: Recorded maximum contraction factors for MgNO (mean).

Formulation	$\rho_{\max}$
Direct + $\mathcal{L}_{\text{dir}}$	0.9831
NPBS + $\mathcal{L}_{\text{bs}}^{\ell_2}$	0.9547
NPBS + $\mathcal{L}_{\text{bs}}^{R_\eta}$	**0.9269**

J.4 Transfer under fixed-PPW scaling

Table 10 evaluates a single MgNO-based metric-matched NPBS model trained at 256^2, with no retraining or fine-tuning on larger grids. The frequency is increased as the grid spacing decreases to retain PPW 12. Thus the table measures transfer across structured-grid resolutions and frequencies, rather than refinement of one fixed-frequency PDE. The 256^2 row is the reference point for this additional transfer evaluation. Table 11 compares all three training formulations in the same resolution study; the matched metric gives the lowest iteration counts at each reported resolution.

Table 10: Metric-matched NPBS resolution transfer with MgNO and PPW 12.

Grid	h (m)	Frequency (Hz)	Avg. Iters	Avg. Time
256×256	2.73	92	126	0.103 s
512×512	1.37	185	220	0.378 s
1024×1024	0.68	370	376	1.870 s
2048×2048	0.34	740	651	11.421 s

Table 11: Iteration counts for the three formulations in the resolution study.

Grid	Direct + $\mathcal{L}_{\mathrm{dir}}$	NPBS + $\mathcal{L}_{\mathrm{bs}}^{\ell_2}$	NPBS + $\mathcal{L}_{\mathrm{bs}}^{R_\eta}$
256×256	342	141	**126**
512×512	483	292	**220**
1024×1024	902	488	**376**

K Limitations

Dimensionality and grid structure. Our experiments cover two-dimensional PDEs and a three-dimensional Helmholtz benchmark on structured Cartesian grids. The current FFT/DST/DCT application of G_η and the evaluated networks do not support general unstructured meshes. Extending them requires a suitable mesh-based reference inverse and compatible architecture. The fixed-PPW transfer tests change frequency with resolution and do not establish mesh-independent convergence for a fixed physical problem. General complex geometries and large-scale three-dimensional cost remain open questions.

Theoretical guarantees. The spectral bounds in Appendix A.4 require $\|G_\eta V_\eta\|_2 \leq q < 1$, whereas the reported experimental norm means exceed one. The learned-iteration convergence and perturbation bounds require the separate uniform residual-reduction condition (28). We have not proved that training enforces this condition, and finite-test contraction statistics cannot certify it globally.

Applicability to PDE classes. NPBS is not suitable for arbitrary PDEs. It relies on a reference–perturbation split $A = L_0 - V$ with an invertible reference operator L_0, so that the Born-preconditioned form $I - G_0 V$ with $G_0 = L_0^{-1}$ is well defined. This structure is natural for the Helmholtz, CDR, and Newton linear systems studied here, where a reference operator captures the main part of the PDE and the remaining coefficient or state dependence is treated as a perturbation. PDEs for which no meaningful or stable reference–perturbation split is available fall outside the present framework.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main contributions and scope. §4 presents the NPBS framework and the metric-matched training objective; §5 contains the experimental process and reported results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses several limitations in Appendix K.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All propositions (Propositions 3.1, 4.1, and C.1) state their assumptions explicitly (e.g., invertibility of L_η, constant-coefficient reference with compatible boundary conditions). Complete proofs are provided immediately after Proposition 3.1, in Appendix A.1, and in Appendix C, respectively.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Network architectures are detailed in Appendix E, dataset descriptions and illustrations in Appendix F, and full training hyperparameters, hardware, and evaluation protocols in Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The Helmholtz datasets (OpenFWI) and the nonlinear PDE setup are publicly available. The CDR dataset is synthetically generated and its generation procedure is fully described in the paper. Code is publicly available at `https://github.com/wangjt23/NeuralPreconditionedBornSeries`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and test details are provided in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We summarize solver performance using average iteration counts and runtimes over the test set. Error bars and formal significance tests are not reported for these comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the compute resources required for our experiments in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This research is mostly theoretical. We do not see obvious social impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No obvious such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cited the original papers and respected their licenses. Specifically, the OpenFWI dataset is licensed under CC BY-NC-SA 4.0, while the Newton Linear Systems dataset, FNO, and encoder-solver are under the MIT license.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The data generation procedure is described in the paper for reproducibility, no external assets are distributed alongside the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects, and therefore IRB approval or equivalent review is not applicable.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were solely used for writing, grammar polishing, and plotting visualization results, which do not impact the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.