Neural Schur Complement Operators for Coarse-to-Fine Neural Operator Transfer

Ansh Tiwari [1]

Abstract

Neural operators are advertised as discretization-flexible, but deploying them from a coarse grid to a finer one routinely loses accuracy that same-grid validation cannot predict. We show that this failure is structural rather than a training artifact: any diagnostic that depends only on the coarse-grid restriction of a fine prediction is blind to the detail component that lives entirely below the coarse resolution. For linear, symmetric positive-definite finite-volume discretizations, the energy-minimizing fine-grid state consistent with a fixed coarse prediction is determined by an algebraic Schur-complement detail solve. We learn this detail map with a Neural Schur Complement Operator, which preserves the coarse prediction exactly by construction and targets the energy-optimal fine correction, and which comes with an additive energy-norm error decomposition and an optional projected conjugate-gradient polishing step. On high-contrast finite-volume Darcy, the method improves over a zero-shot lift by 1.67 to $6.21\times$, recovering 84.2 to 103.1% of the oracle Schur gain, and matches direct fine-grid training at roughly one-tenth the fine-label budget. We also report the limits honestly: with real coarse predictors the achievable gain is bounded by source quality, and a model trained on one permeability morphology does not transfer to channelized out-of-distribution media without retraining.

1. Introduction

Neural operators for partial differential equations (Kovachki et al., 2023; Li et al., 2021) advertise discretization-flexibility: a single set of parameters should produce useful predictions across grids, meshes, and basis families through mechanisms ranging from spectral mode truncation to anti-aliased convolutions and graph encodings. In practice, deploying a model from a coarse grid to a finer target grid loses accuracy that same-grid validation does not predict, because validation on the coarse grid cannot see the part of the error that lives below the coarse resolution. A natural label-free remedy is to monitor a commutator gap between restriction and prediction, but this remedy is unreliable for the structural reason we develop below; in a PDEBench (Takamoto et al., 2022) pilot it fails three of four pre-registered guarantees, and we defer that backstory to Sections D.1 and D.3.

The reason for that failure is structural. Any quantity built from $RT_f\,L$ depends on T_f only modulo $\ker R$: for any map K with image in $\ker R$, $R(T_f + K)L = RT_f\,L$. We formalize this as the Nullspace No-Go theorem (Theorem 3.1). The same blindness afflicts residual-only correction in $\ker R$ without an algebraic supervision signal: on our hard-Darcy benchmark, residual-only training is *worse* than zero-shot lift in 8 of 9 cells (Section D.2). The problem is not the loss but the geometry.

This paper tests one structural hypothesis: that coarse-to-fine transfer should be posed not as an unconstrained regression onto fine fields, but as the recovery of the unique fine-grid detail that the coarse prediction leaves undetermined. The rest of the paper follows a single arc — a structural impossibility result, the algebraic Schur target it implies, a network that meets that target by construction, controlled evidence that the target is learnable, and an honest account of where real coarse predictors and out-of-distribution media break it.

The geometry has a concrete physical meaning for the high-contrast elliptic problems we study. A coarse cell averages the fine-grid pressure over a block of fine cells, so the coarse prediction fixes only the block means and says nothing about how the field varies inside each block. In a high-contrast permeability medium, flux concentrates in thin conductive channels and bends sharply around low-permeability inclusions, and much of that variation is precisely the sub-block detail that block-mean restriction discards. A correction that only adjusts coarse-visible degrees of freedom therefore cannot, even in principle, restore the within-block structure that the fine solve resolves; recovering it requires an operator

[1]California Institute of Technology, Pasadena, CA, USA. Correspondence to: Ansh Tiwari <ansh@caltech.edu>.

Proceedings of the AI4Physics Workshop at the 43rd International Conference on Machine Learning (AI4Physics@ICML 2026), Seoul, South Korea. 2026. Copyright 2026 by the author(s).

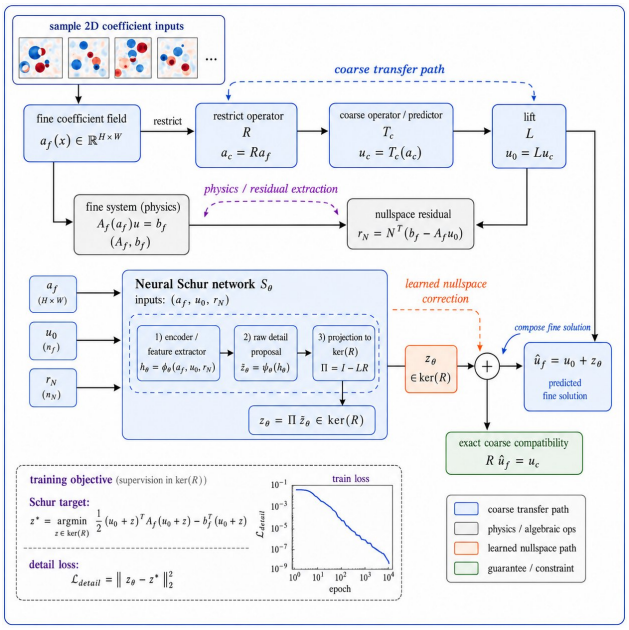


Figure 1. **Neural Schur Complement Operator.** The coarse transfer path (blue) restricts the fine coefficient field ($a_c = Ra_f$), applies the coarse operator T_c ($u_c = T_c(a_c)$), and lifts back to the fine grid ($u_0 = Lu_c$). The fine physics $A_f(a_f)\,u = b_f$ yields the nullspace residual $r_N = N^\mathsf{T}(b_f - A_f u_0)$, equal to the projected fine-grid residual $r_\Pi = \Pi(b_f - A_f u_0)$ used as the network input channel. The Neural Schur network S_θ takes (a_f, u_0, r_N), forms a raw detail proposal, and projects it onto $\ker R$ with $\Pi = I - LR$, giving $z_\theta \in \ker R$. The fine prediction $\hat{u_f} = u_0 + z_\theta$ satisfies $R\hat{u_f} = u_c$ exactly. Training supervises z_θ against the Schur target $z^\star = \arg\min_{z\in\ker R} \frac{1}{2}(u_0+z)^\top A_f(u_0+z) - b_f^\top(u_0+z)$ with the detail loss $\|z_\theta - z^\star\|^2$.

that acts in the space of fields the restriction maps to zero.

For an SPD fine-grid operator A_f and a coarse prediction u_c, the unique field minimizing the fine-grid energy subject to $Ru = u_c$ is the *Schur lift* $u^\star(u_c) = Lu_c + Nz^\star$, where $z^\star$ solves the Schur detail equation $S_f z^\star = N^\mathsf{T}(b_f - A_f Lu_c)$ in any basis N of $\ker R$. In the implementation we do not materialize N: the network predicts a fine-grid proposal and we project it with $\Pi = I - LR$, giving the *Neural Schur Complement Operator* $\hat{u_f} = Lu_c + \Pi\, S_\theta(a_f, u_0, r_\Pi)$ with $r_\Pi = \Pi(b_f - A_f u_0)$, sketched in Figure 1. By construction the coarse prediction is preserved exactly (Theorem 4.1); the network targets the energy-optimal $z^\star$ (Theorem 4.2); the fine-grid energy-norm error decomposes additively into source quality, detail-approximation error, and transfer geometry (Theorem 4.3); the projected-residual training objective is provably equivalent to direct supervision under preconditioning (Theorem 4.4); and a projected conjugate-gradient polish inherits a $((\sqrt{\kappa}-1)/(\sqrt{\kappa}+1))^K$ contraction (Theorem 4.5).

Empirically, on hard finite-volume Darcy with six orders of magnitude permeability contrast and 9 source-noise / coarse-resolution cells across 5 seeds, Neural Schur improves over zero-shot lift by a factor of 1.67 to 6.21 and recovers 84.2 to 103.1% of the algebraic-Schur oracle gain. Trained on as few as 10 fine labels it reaches an error that a direct fine-grid FNO does not reach with 100, and in a fine-solver-validated pressure-constrained well-placement loop it improves surrogate design ranking over zero-shot lift using a single fine solve per design, though a non-negligible absolute gap to the fine optimum remains. We report the trained model's failure mode honestly: out-of-distribution channelized permeability fields break a S_θ trained on disc-inclusion fields (Section C.3).

Contributions.

- **A structural no-go result.** Diagnostics built from a coarse-grid restriction of the fine prediction — commutator and coarse-consistency objectives — are provably blind to the detail subspace $\ker R$, so they cannot certify cross-grid transfer (Theorem 3.1).
- **A Schur-compatible method.** The Neural Schur Complement Operator learns only the detail correction and preserves the coarse prediction exactly by construction (Theorem 4.1), targeting the unique energy-minimizing Schur lift (Theorem 4.2).
- **Supporting theory.** An additive energy-norm error decomposition (Theorem 4.3), a preconditioned-residual training equivalence (Theorem 4.4), and a projected-CG warm-start convergence rate for an optional polishing step (Theorem 4.5).
- **Controlled and honestly-limited empirical evidence.** On high-contrast finite-volume Darcy the method gives 1.67 to 6.21× over zero-shot lift at 84.2 to 103.1% oracle recovery and matches direct fine training at one-tenth the fine-label budget; we also document where it is limited, namely source quality with real coarse predictors and failure on channelized out-of-distribution media (Section C.3).

2. Related Work

Discretization-flexible neural operators. A first line of work seeks a single neural operator usable across discretizations: spectral mode truncation in the Fourier neural operator (Li et al., 2021), anti-aliased convolutions in convolutional and representation-equivalent neural operators (Raonic et al., 2023; Bartolucci et al., 2023), point-cloud and graph encodings for resolution- and mesh-independent variants (Li et al., 2023; Bahmani et al., 2025), and multigrid pyramids on top of spectral modes (Kossaifi et al., 2024). Each mechanism guarantees parameter sharing across grids, none guarantees algebraic consistency between fine and coarse predictions: weight sharing is not commutation with arbitrary FE/FV transfers, and the classical com-

mutativity diagrams in finite-element exterior calculus use prolongations that differ from spectral mode-padding by exactly the $\ker R$ component our construction controls.

Hybrid neural-iterative solvers and residual correction. The closest prior work combines a learned operator with a classical iterative solver or a learned residual: hybrid neural-iterative solvers alternate a learned preconditioner with classical sweeps (Zhang et al., 2022), physics-informed multigrid neural operators use a hierarchy with a physics-informed loss (Raissi et al., 2019), learned multigrid prolongation generalises the inter-grid operator (Greenfeld et al., 2019), and residual-correction networks train a learned residual on top of a learned operator (Cao et al., 2023). None derives the learning target from the algebraic Schur complement, none enforces $R\hat{u_f} = u_c$ by construction, and we show in Section D.2 that residual-only correction in $\ker R$ without an algebraic target diverges in 8 of 9 hard-Darcy cells.

Classical Schur, domain decomposition, and multigrid. The Schur-complement formulation we exploit is classical in domain decomposition, algebraic multigrid, and constrained quadratic programming with KKT block elimination (Trottenberg et al., 2001; Saad, 2003). Classical Schur solves one instance to a tolerance; the contribution here is to amortize the Schur solver S_θ across a coefficient distribution, reducing per-instance inference to one network pass and optionally a small number of projected-CG steps. In short, prior neural-operator transfer methods guarantee parameter sharing across grids but not algebraic consistency, and prior hybrid/residual solvers add a learned correction without a Schur-derived target or an exact $R\hat{u_f} = u_c$ constraint. Our method is the only one that simultaneously preserves the coarse prediction exactly, learns the algebraic Schur detail map, transfers label-efficiently to the fine grid, and admits a CG-polishable certified rate.

3. Setting and the Quotient/Detail Decomposition

Notation. We write R for the block-mean restriction from the fine grid to the coarse grid and L for the matching block-replicate prolongation, so that $RL = I$ on nested grids. The projector $\Pi = I - LR$ removes the coarse-visible component of a fine-grid field and leaves only the detail component in $\ker R$. A coarse predictor T_c produces a coarse state u_c; its lifted fine-grid state is $u_0 = Lu_c$, and the projected fine-grid residual of this guess is $r_\Pi = \Pi(b_f - A_f u_0) \in \ker R$, stored as a fine-grid field. The Schur detail target is denoted $z^\star$ and the Schur complement is $S_f = N^\mathsf{T} A_f N$, where N denotes any basis for $\ker R$ used only in the theory; the basis-coordinate residual $r_N = N^\mathsf{T}(b_f - A_f u_0)$ is the right-hand side of the coordinate Schur equation $S_f z = r_N$. In the implementation we never materialize a dense N and instead apply Π directly to the network output. The fine system operator is A_f and the fine solution is u_f.

3.1. Forward problem

We work with the parametric elliptic boundary-value problem $-\nabla \cdot (a(x)\nabla u(x)) = f(x)$ on $\Omega = [0,1]^d$ with zero Dirichlet boundary, discretized on a regular finite-volume grid of size N_f along each axis. Cell-centered face-averaging produces an SPD linear system

$$A_f(a)\, u_f \;=\; b_f, \tag{1}$$

where $A_f(a) \in \mathbb{R}^{n_f \times n_f}$ is sparse with 5- (resp. 7-) point stencil in $d = 2$ (resp. $d = 3$). Our principal benchmark is hard 2D Darcy with permeability $\log_{10} a \in [-3, 3]$ produced by a Gaussian random field plus randomly placed disc inclusions; full generator parameters are in Section B.

3.2. Coarse-fine transfer and the algebraic decomposition

Let $R : \mathbb{R}^{n_f} \to \mathbb{R}^{n_c}$ be the block-mean restriction onto a nested coarse grid of size N_c with refinement ratio $r = N_f/N_c$, and let $L : \mathbb{R}^{n_c} \to \mathbb{R}^{n_f}$ be the matching block-replicate prolongation (Trottenberg et al., 2001; Brandt, 1977). Then $RL = I_c$ exactly, and the operator $\Pi = I_f - LR$ is the orthogonal projector onto the kernel $\ker R$. Choose $N \in \mathbb{R}^{n_f \times (n_f - n_c)}$ whose columns span $\ker R$ so that $RN = 0$ and $[L \mid N]$ is full column rank. Every fine field admits a unique algebraic decomposition

$$u_f \;=\; L(Ru_f) \;+\; (I_f - LR)u_f \;\in\; \mathrm{Range}(L) \oplus \ker R. \tag{2}$$

The summands are the *quotient* (coarse-visible) and the *detail*; on nested regular grids the decomposition is exact. The transfer geometry constants $RL = I$ and $RN = 0$ become approximate on unstructured meshes; we discuss this in Section 7.

3.3. Why commutator-only diagnostics fail

Suppose one searches for a label-free predictor of cross-discretization error of the form $g(RT_{\theta,f}L, T_{\theta,c})$ for some function g. The following observation rules out any such construction.

Theorem 3.1 (Nullspace No-Go)**.** *Let R be non-injective and let $T_f, K : X \to \mathbb{R}^{n_f}$ be arbitrary maps with $\mathrm{Range}(K) \subseteq \ker R$. Then for every $x \in X$, $R(T_f + K)x = RT_f x$. Consequently, any criterion of the form $g(RT_f L, T_c)$ depends on T_f only modulo $\ker R$.*

The proof, by linearity of R and $RK = 0$, is one line and is given in Section A.1; Figure 9 in Section D.1 illustrates the

phenomenon empirically with two fine fields whose restrictions are visually identical but whose per-pixel detail magnitudes differ markedly. Empirically, Theorem 3.1 explains why the commutator-gap diagnostic fails on PDEBench Darcy and Burgers (Section D.1) and why residual-only correction in $\ker R$ diverges on hard Darcy (Section D.2). It also tells us what to do next: any constructive method must build the coarse-consistency in by construction and learn the $\ker R$ component against an algebraic supervision signal, which is the topic of Section 4.

4. Neural Schur Complement Operators

4.1. Architecture

The Neural Schur Complement Operator parameterizes a fine-grid prediction by lifting a coarse prediction and adding a network-predicted detail correction in $\ker R$:

$$\boxed{\hat{u_f} \;=\; Lu_c \;+\; \Pi\, S_\theta(a_f, u_0, r_\Pi)} \tag{3}$$

with $u_0 = Lu_c$ and the projected fine-grid residual $r_\Pi = \Pi(b_f - A_f u_0) \in \ker R$, represented as a fine-grid field. Here u_c is a coarse prediction (from a frozen coarse operator or from synthetic noise around the restricted ground truth, depending on the experiment), u_0 is the lifted coarse guess, and r_Π is the projected residual of u_0 on $\ker R$. The projector $\Pi = I_f - LR$ is applied after the network so that $z_\theta := \Pi S_\theta(\cdot)$ satisfies $Rz_\theta = 0$ to machine precision regardless of the network output.

We instantiate S_θ as a Fourier neural operator backbone (Li et al., 2021) (width 32, 16×16 Fourier modes, 4 layers, $\sim$4.2M parameters) with input channels (a_f, u_0, r_Π, x, y). One- and three-dimensional variants and the load-bearing input normalization $a \mapsto \log_{10} \max(a, 10^{-6})/3$ and $r_\Pi \mapsto h^2 r_\Pi$ are specified in Section B; the importance of the normalization is quantified in Section 6.

The Fourier backbone is only a parameterization of the Schur detail map; the $\ker R$ constraint is enforced by the final projection Π, not by selecting high-frequency Fourier modes. We chose the mode count by validation at the anchor cell, and the "lower modes only" ablation is statistically comparable to the default (Section 6), so the empirical gain should be attributed to the Schur-compatible parameterization rather than to a particular spectral cutoff.

4.2. Exact compatibility and the Schur target

The architecture admits two immediate algebraic guarantees.

Theorem 4.1 (Exact coarse compatibility)**.** *For the architecture in Equation (3) and any choice of R, L with $RL = I$, the prediction satisfies $R\hat{u_f} = u_c$ for every parameter θ, every input a_f, and every coarse prediction u_c.*

Theorem 4.2 (Schur optimality)**.** *Assume $A_f(a)$ is symmetric positive definite. The unique minimizer of the constrained energy $\min_u \frac{1}{2} u^\mathsf{T} A_f u - b_f^\mathsf{T} u$ subject to $Ru = u_c$ equals $u^\star(u_c) = Lu_c + N\, z^\star$, where*

$$z^\star \;=\; (N^\mathsf{T} A_f N)^{-1}\, N^\mathsf{T}(b_f - A_f L u_c) \;=\; S_f^{-1}\, r_N, \tag{4}$$

and $S_f := N^\mathsf{T} A_f N$ is the Schur complement (Saad, 2003; Trottenberg et al., 2001).

Together, Theorems 4.1 and 4.2 identify a unique algebraic supervision signal: the network should predict $z_\theta \approx z^\star$, where $z^\star$ is computed once per training sample by a sparse KKT block solve on the full fine grid (Section B). The proofs are in Sections A.2 and A.3.

4.3. Energy-norm error decomposition

Theorem 4.3 (Energy-norm error decomposition)**.** *Let $u_f^\star$ denote the true fine-grid solution of Equation (1) and $\hat{u_f}$ the prediction of Equation (3). Under the assumptions of Theorem 4.2 there exist constants $C_1, C_2, C_3 \geq 0$ depending only on the transfer pair (R, L, N) and the SPD operator A_f such that*

$$\begin{aligned} \|u_f^\star - \hat{u_f}\|_{A_f} \;&\leq C_1\, \|Ru_f^\star - u_c\|_{A_c} \\ &\quad +\; C_2\, \|z^\star - z_\theta\|_{S_f} \;+\; C_3\, \eta_{R,L}. \end{aligned} \tag{5}$$

The three summands correspond to (I) source quality, (II) detail approximation, and (III) transfer geometry: (I) is the responsibility of the coarse predictor T_c, (II) the responsibility of S_θ (capacity, training data, loss), and (III) measures the transfer geometry, vanishing on nested regular grids when $RL = I$ and $RN = 0$. We verify each term empirically: (I) by varying T_c in Section 5.3, (II) by varying training data and capacity in Sections 5.4 and 6, and (III) by sweeping transfer families in Section C.1. The proof is in Section A.4.

4.4. Training regimes

Theorem 4.2 gives a direct supervised loss $\mathcal{L}_{\sup}(\theta) = \mathbb{E}\, \|z_\theta - z^\star\|_2^2$ with $z^\star$ from Equation (4). A label-free alternative is the projected-residual loss $\mathcal{L}_M(\theta) = \mathbb{E}\, \|S_f z_\theta - r_N\|_M^2$ for a preconditioner $M \succeq 0$, in the same family as physics-informed losses (Raissi et al., 2019) but defined on the kernel subspace. The two are tightly related.

Theorem 4.4 (Preconditioned residual equivalence)**.** *Let M be a symmetric positive-definite matrix satisfying $\alpha\, S_f^{-1} \preceq M \preceq \beta\, S_f^{-1}$ for some $0 < \alpha \leq \beta$. Then for every z, $\alpha\, \|z - z^\star\|_{S_f}^2 \;\leq\; \mathcal{L}_M(z) \;\leq\; \beta\, \|z - z^\star\|_{S_f}^2$.*

In particular, with $M = I$ the bound becomes $\kappa(S_f)$-controlled. On hard Darcy $\kappa(S_f) \sim 10^6$, which explains why the unpreconditioned label-free loss stalls at zero-shot

in our experiments. In our implementation the residual-loss ablation uses $M = \mathrm{diag}(S_f)^{-1}$ (a Jacobi preconditioner, approximated by $\mathrm{diag}(S_f)_i \approx \mathrm{diag}(A_f)_i\,(1 - 1/r^2)$) by default, with an alternative K-step dual/implicit-CG preconditioner; the default supervised loss $\mathcal{L}_{\mathrm{sup}}$ uses no preconditioner. Theorem 4.4 certifies that whenever M satisfies $\alpha S_f^{-1} \preceq M \preceq \beta S_f^{-1}$, minimizing $\mathcal{L}_M$ minimizes $\|z - z^\star\|_{S_f}$ up to the factor β/α; we report the residual-loss ablation with this specific Jacobi M and do not claim it makes β/α small in practice. The proof is in Section A.5.

By default we use $\mathcal{L}_{\mathrm{sup}}$ (supervision against $z^\star$) for the controlled experiments in Section 5; the label-free $\mathcal{L}_M$ is studied as an ablation in Section C.1, and the few-label calibrated regime is discussed in Section 5.4.

Energy norm versus L^2 loss. Theorems 4.3 and 4.4 are energy/Schur-norm statements. Our default supervised experiments minimize a normalized Euclidean loss on the precomputed detail target $z^\star$ because this is stable and directly comparable to neural-operator baselines, and we likewise report relative L^2 error throughout Section 5. This choice does not by itself provide a certified energy-norm training guarantee; an energy-weighted loss or the preconditioned residual loss $\mathcal{L}_M$ would be needed for that interpretation.

4.5. Optional projected-CG polish

Theorems 4.1 and 4.2 place the Neural Schur output on the affine subspace $\{u : Ru = u_c\}$ but do not, on their own, guarantee a particular distance from the constrained optimum. A classical projected-CG iteration (Hestenes & Stiefel, 1952; Saad, 2003) on the constrained system $\Pi^\mathsf{T} A_f \Pi\, z = \Pi^\mathsf{T}(b_f - A_f u_0)$ provides exactly this guarantee.

Theorem 4.5 (Neural Schur + projected CG)**.** *Let $z_0 = S_\theta(a_f, u_0, r_\Pi)$ be the network output and z_K the iterate produced by K steps of projected conjugate gradient on $\Pi^\mathsf{T} A_f \Pi$ starting from z_0. Then with κ the condition number of S_f,*

$$\|z_K - z^\star\|_{S_f} \;\le\; 2\Big(\tfrac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^K \|z_0 - z^\star\|_{S_f}. \qquad (6)$$

For a well-trained network the warm-start factor $\|z_0 - z^\star\|_{S_f}/\|z^\star\|_{S_f}$ is small, so the leading constant 2 is replaced by this ratio in practice; we measure both quantities in Section 5.5.

The proof, an application of the Chebyshev energy-norm bound for CG, is given in Section A.6.

5. Experiments

Setup. All training and evaluation use hard finite-volume Darcy on $N_f = 128$ unless stated otherwise; full generator parameters are in Section B. Unless otherwise stated, each controlled-matrix cell reports 5 seeds (mean $\pm$ standard deviation); statistical tests are deferred to Section E. The principal baselines are zero-shot lift ($\hat{u_f} = LT_c(Ra_f)$), the Schur oracle (KKT block solve; Theorem 4.2), full-fine-grid FNO with full labels, multi-resolution FNO, and residual-only correction without $\ker R$ projection. Hardware: $8\times$ H200 for the controlled matrix; $1\times$ A100 for the application loop; Lambda for the topology benchmarks. The controlled matrix uses a synthetic coarse state that isolates the detail-learning problem from coarse-predictor quality, whereas the real-T_c experiments (Section 5.3) substitute trained coarse predictors to test source-quality sensitivity.

Physical setting. The benchmark solves steady single-phase Darcy flow, $-\nabla \cdot (a(x)\,\nabla p) = f$, for pressure p in a medium whose permeability $a(x)$ spans six orders of magnitude between low-permeability matrix and high-permeability inclusions. A cell-centered finite-volume discretization turns this elliptic problem into the SPD linear system $A_f u = b_f$, where the entries of A_f are harmonic face transmissibilities that couple neighbouring cells. Block-mean restriction R to a coarse grid averages the pressure over each coarse block, which discards the sub-block flux structure that high-contrast inclusions induce: two fine fields that differ only inside a coarse block have identical coarse restrictions, so the lost detail lives entirely in $\ker R$. Recovering that sub-block detail is exactly the Schur complement solve our method amortizes, and it is what same-grid coarse validation cannot see.

5.1. Hard FV Darcy: benchmark and headroom

A coarse-to-fine transfer benchmark is informative only when the target discretization carries a meaningful detail component. PDEBench (Takamoto et al., 2022) Darcy at $64 \to 128$ has detail headroom $H_R(L) := \|u_f - LRu_f\|/\|u_f\|$ of approximately 0.1%; our hard Darcy generator produces fields with mean $H_R(L)$ ranging from 7.3% at $N_c = 64$ to 28.8% at $N_c = 16$ (Table 8, with the headline visualisations deferred to Figure 5 in the appendix). The same-grid PDEBench zero-headroom is the structural reason an early commutator/residual prototype on PDEBench Darcy could not improve over zero-shot (Section D.3).

5.2. Headline controlled source-quality matrix

The synthetic-u_c experiments here are a controlled source-quality study, not a deployment model. By perturbing Ru_f directly we isolate whether the learned Schur detail map S_θ can recover the fine-scale component when the coarse state is known up to a prescribed error, decoupling detail learning from coarse-predictor quality. Section 5.3 then replaces this controlled source with trained coarse operators and exposes the expected source-quality dependence. On

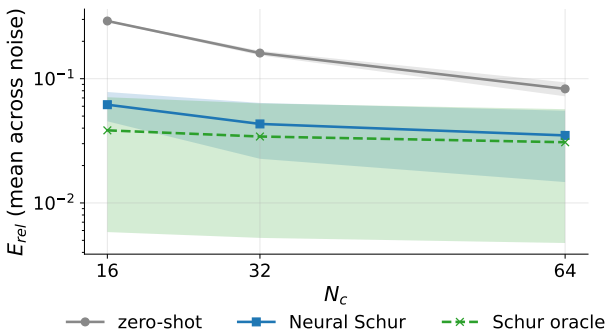


Figure 2. **Relative error vs coarse resolution** N_c (F021), averaged across noise levels $\{0.01, 0.05, 0.10\}$. Solid lines are means; shaded bands are ± 1 standard deviation across noise. Log y-axis. Neural Schur tracks the Schur oracle and stays well below zero-shot at every N_c.

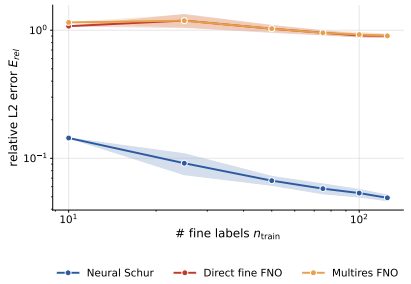


Figure 3. **Label efficiency of Neural Schur** (F113). Test error vs number of fine labels n_{train} for Neural Schur (supervised on $z^\star$), full-fine FNO, and multi-resolution FNO; log-log axes. Zero-shot lift error is shown as a horizontal reference line. At $n_{\text{train}} = 10$, Neural Schur reaches an error that direct fine FNO does not match at $n_{\text{train}} = 100$.

a 3×3 grid of (N_c, noise) cells with $N_c \in \{16, 32, 64\}$ and noise $\in \{0.01, 0.05, 0.10\}$, 5 seeds per cell, $n = 500$ samples per cell ($350/75/75$ train/val/test split), Neural Schur recovers 84.2–103.1% of the algebraic-Schur oracle gain and improves over zero-shot lift by a factor of 1.67 to 6.21 (Table 1). The strongest absolute gain is at small N_c and low source-noise; the strongest relative-to-oracle recovery is at high source-noise, where the achievable oracle gap is small. Figure 2 plots the seed-averaged error against N_c with shaded bands across noise levels: Neural Schur sits between zero-shot and the algebraic oracle, and the gap closes as N_c grows. Figure 5b (N_c=32, noise$= 0.05$) confirms that gains are not driven by a few outliers: every test sample lies below the diagonal. Heatmap, per-seed numbers, and bootstrap intervals are deferred to Figure 5a and Sections C.2 and E.

5.3. Real coarse T_c: source quality matters

The headline matrix uses synthetic noise around Ru_f as a stand-in for the coarse prediction u_c. Replacing this stand-in with the output of a trained coarse operator (FNO, DeepONet, or CNO (Li et al., 2021; Lu et al., 2021; Raonic et al., 2023), all with log-a normalized inputs) makes the source-quality term (I) of Theorem 4.3 the dominant limitation (Table 14, full grid in Section C.9). With FNO and CNO coarse operators the oracle gap stays positive at all N_c and Neural Schur recovers it almost entirely: relative-L^2 error drops from $0.39/0.43$ (zero-shot) to $0.30/0.36$ at N_c=16 and from $0.19/0.22$ to $0.17/0.21$ at N_c=64, tracking the corresponding Schur oracle. The DeepONet coarse model, by contrast, leaves little or no usable oracle gap: its mean Schur-oracle error is no better than its zero-shot lift at every N_c, so the oracle-recovery ratio is undefined (reported as NA in Table 14) rather than a meaningful percentage. These results are noisier than the synthetic-u_c matrix and should be read as evidence for the error decomposition of Theorem 4.3, not as architecture-independent deployment robustness: improving the coarse operator pays for itself end-to-end, but only when the coarse predictor is good enough to leave a positive oracle gap.

5.4. Label efficiency

A practical motivation for separating the coarse and detail learning problems is that the detail map S_θ should require fewer fine labels than a direct fine-grid solution operator. We test this by training Neural Schur, a direct full-fine FNO (Li et al., 2021), and a multi-resolution FNO (Kossaifi et al., 2024) at $n_{\text{train}} \in \{10, 25, 50, 75, 100, 125\}$ fine labels each, with the same backbone capacity. Figure 3 and Table 2 report the result.

Table 2 shows that at $n_{\text{train}} = 10$, Neural Schur reaches error 0.144 ± 0.048, which the full-fine FNO does not reach at $n_{\text{train}} = 100$ (where it is still at 0.909, more than $6\times$ worse). The gap is consistent with Theorem 4.3: Neural Schur is spending its capacity on a $\ker R$-only target rather than on the full solution, and the algebraic supervision $z^\star$ is far better specified than the unconstrained full-grid target.

5.5. Runtime / error Pareto with CG polish

Theorem 4.5 predicts that polishing the network output with K projected-CG steps inherits a $((\sqrt{\kappa}-1)/(\sqrt{\kappa}+1))^K$ contraction. We measure this directly. CG-from-zero on the ill-conditioned fine operator makes essentially no progress at small K, while Neural Schur already sits well below the zero-shot baseline at $K = 0$, and a small number of projected-CG steps on top approaches the Schur oracle. The runtime/error Pareto curve is in Figure 8; the per-method runtime, parameter, and memory budget is in Table 13 (both in Section C.8).

Table 1. **Headline matrix on hard FV Darcy** (T020). Per-cell summary, 5 seeds. E_{zero} is zero-shot lift, E_{neural} is Neural Schur, E_{oracle} is the algebraic Schur oracle. Recovery $= 100\,(E_{\text{zero}} - E_{\text{neural}})/(E_{\text{zero}} - E_{\text{oracle}})\%$; improvement $= 100\,(E_{\text{zero}} - E_{\text{neural}})/E_{\text{zero}}\%$. Recovery is computed in relative L^2 error, whereas the Schur oracle is energy-optimal under the fixed coarse constraint; an uncapped recovery slightly above 100% (here a single cell at 103.1%) can occur when the learned predictor is marginally closer in L^2 than the energy-optimal constrained lift, especially when the oracle gap is small. We read such cases as oracle parity, not as a violation of the Schur optimality theorem (Theorem 4.2).

N_c	noise	rec. %	rec. std	E_{zero}	E_{neural}	E_{oracle}	seeds	improv. %
16	0.0100	85.3504	0.6774	0.2881	0.0483	0.0072	5	83.2186
16	0.0500	91.6686	0.3966	0.2901	0.0570	0.0358	5	80.3661
16	0.1000	96.4965	0.5231	0.2960	0.0800	0.0722	5	72.9572
32	0.0100	87.5069	0.9450	0.1553	0.0250	0.0064	5	83.9002
32	0.0500	94.3994	0.5004	0.1589	0.0392	0.0321	5	75.3500
32	0.1000	98.7770	0.4142	0.1683	0.0656	0.0643	5	61.0426
64	0.0100	84.1751	1.2231	0.0737	0.0165	0.0058	5	77.5848
64	0.0500	94.4207	0.8702	0.0802	0.0317	0.0289	5	60.4309
64	0.1000	103.0822	0.7679	0.0946	0.0566	0.0578	5	40.1414

Table 2. **Label-budget table.** Mean±std test error over up to 7 seeds at $N_c = 32$, noise$= 0.05$ (1 seed at n_{train}=10, 6–7 seeds otherwise). Methods: Neural Schur (NS), full-fine FNO (FNO), multi-resolution FNO (mFNO). The full per-seed breakdown is in Section C.2.

n_{train}	NS	FNO	mFNO
10	0.144±.048	1.076±.396	1.151±.452
25	0.092±.017	1.190±.138	1.188±.127
50	0.067±.005	1.027±.065	1.029±.050
75	0.058±.005	0.955±.036	0.954±.033
100	0.054±.004	0.909±.024	0.925±.038
125	0.049±.003	0.903±.027	0.904±.024

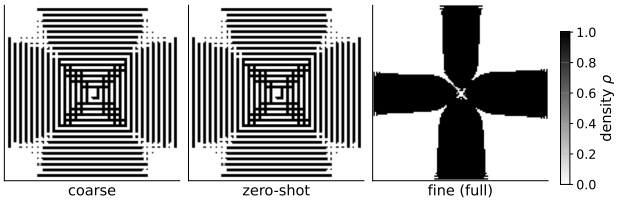


Figure 4. **A4 thermal designs** (F231). Final material density fields from SIMP under the coarse surrogate, its zero-shot lift, and the fine reference; greyscale density from 0 (void) to 1 (solid). All rendered designs converge to the same branching topology family. Fine-validated temperature fields and the SIMP optimization trajectory are in Figures 6 and 7 (Section C.4).

5.6. Application stress test: thermal topology optimization

A forward-error claim is informative only insofar as it survives a real decision loop. We include a fine-solver-validated thermal topology-optimization task as an application stress test for coarse-to-fine surrogate design. This A4 experiment isolates the coarse-vs-fine SIMP (Sigmund, 2001; Bendsøe & Sigmund, 2003) design gap; a Neural Schur A4 surrogate-design loop was not recorded and is not claimed here. Three forward models drive the design loop here—a coarse-only surrogate, its zero-shot block-replicate lift, and a fine reference solver; every final design is then validated on the 128×128 fine FV solver, and we report the fine compliance, peak temperature, and volume-fraction deviation. The full multi-seed aggregation is in Table 3; the final designs, fine-validated temperature fields, and optimization trajectories are in Figure 4 and, for the validated temperature fields and SIMP optimization trajectory, in Figures 6 and 7 (Section C.4). A cross-suite summary including A1 well placement and A3 CO_2 pressure management is in Table 4; per-application detail is in Section C.4. This A4 study isolates the coarse-vs-fine SIMP design gap; a Neural Schur surrogate-design variant for A4 is out of scope of the rendered figures and is deferred to future work.

Metric conventions. Each application optimizes its own physical objective in its own units, so values are comparable only *within* an application, never across rows of Table 4. For all reported objectives lower is better. We report the across-seed (or across-field) *mean* unless a quantity is explicitly marked as a median; for the well-placement objective J_f the per-field distribution is heavy-tailed, so we report the median in Table 4 and give both the mean (6.27) and the median (0.0034) in Table 11. The "fine reference" column is the design found by running the optimizer on the full fine solver, and "fine solves" counts fine-grid linear solves consumed by the surrogate.

The A4 thermal task is reported separately in Table 3 as a coarse-vs-fine stress test without a Neural Schur design loop; the out-of-distribution SPE10 result is in Table 10.

5.7. Out-of-distribution external data

On out-of-distribution external SPE10-style channelized permeability, a S_θ trained on the disc-inclusion generator fails: at $N_c = 16$ the oracle-gain recovery is -144.6% (worse than zero-shot; Section C.3). We keep this as a stated limitation: S_θ does not transfer across coefficient distributions

Table 3. **A4 thermal topology, multi-seed aggregation** (T230). Across-seed mean and std of validated fine compliance J_f, mean peak temperature, volume-fraction deviation, mean wallclock, and seed count, for the coarse surrogate, its zero-shot lift, and the fine reference. In this run the coarse surrogate and its zero-shot lift coincide; a Neural Schur A4 variant is deferred (see text).

method	mean_J_f	std_J_f	median_J_f	mean_peak_T	mean_volume_violation	mean_wallclock_s	n_seeds
coarse	7.354e-05	9.441e-07	7.374e-05	0.00212	5.836e-06	0.294	4
zero-shot	7.354e-05	9.441e-07	7.374e-05	0.00212	5.836e-06	0.2872	4
fine ref.	6.846e-06	5.778e-09	6.845e-06	0.000208	1.062e-05	1.802	3

Table 4. Application outcomes for the two tasks where a Neural Schur surrogate was evaluated (A1 well placement, A3 CO_2 pressure). Metrics are task-specific and comparable only within each row; lower is better. The final column reports fine-grid solves used by the Neural Schur surrogate-design protocol, not by the fine-reference optimizer. Full per-application statistics are in Tables 11 and 12.

		objective (lower better)				fine
app	metric	coarse	zero-shot	NS	fine	solves
A1 well place.	regret[m]	0.023	0.023	0.0034	−0.0008	1
A3 CO_2	max press.	3.0e−5	3.0e−5	2.6e−5	4.5e−5	42

[m] A1 J_f is the per-seed median; the heavy-tailed per-seed *mean* is 6.27 (Table 11).

Table 5. **Decisive structural ablations** (T100), 10 seeds at $N_c = 32$, noise$= 0.05$; the full 14-variant table is Table 7. E_{target} is relative-L^2 error; E_{compat} is the residual of $Ru = u_c$; recovery is $100(E_{\text{zero}} - E_{\text{target}})/(E_{\text{zero}} - E_{\text{oracle}})\%$.

variant	E_{target}	E_{compat}	rec. %
no r_Π input	0.0546	0.0000	81.9
full	0.0586	0.0000	78.2
no projection	0.0611	0.0229	75.9
no u_0 input	0.1510	0.0000	−7.1
no log-a norm.	0.2395	0.0000	−88.9
direct FNO (eq. param)	1.2119	1.1349	−986.9

without retraining, and out-of-distribution channelized media are exactly the setting where its training prior is wrong.

6. Ablations

We attribute the headline gain to four structural choices, each falsifiable by a single ablation. The compact ablation in Table 5 reports the decisive variants at $N_c = 32$, noise $= 0.05$, 10 seeds; the full 14-variant ablation with per-sample ECDFs is in Section C.1.

Table 5 supports four interpretations, with r_Π as an explicit caveat. *Projection is essential.* Removing the kernel projection Π violates Theorem 4.1: the compatibility error E_{compat} rises from 3×10^{-8} to 0.023 (six orders of magnitude), and target error degrades by approximately 5%. *The lifted coarse state u_0 is essential.* Removing it raises target error from 0.059 to 0.151 and drives recovery negative (−7.1%): the network cannot localize the correction within $\ker R$ without the coarse anchor. *Log-a normalization is load-bearing.* Without it, recovery collapses to −88.9%, confirming that the six-orders-of-magnitude permeability span requires explicit normalization. *Schur structure is the gain, not capacity.* An equal-parameter direct FNO predicting u_f from a_f gets 1.21 at the same parameter budget, approximately 20× worse than full Neural Schur. The decisive structural choices are therefore the projection, the lifted coarse state u_0, log-a normalization, and the Schur-compatible parameterization. The ablation does *not* support the stronger claim that r_Π is essential in the supervised $z^\star$-training regime: removing the r_Π input gives statistically comparable, and in this table slightly lower, target error than the full model (0.055 vs. 0.059). We keep r_Π because it is the fine-grid image of the algebraically natural Schur right-hand side, but we do not claim that this supervised ablation isolates it as the source of the gain.

7. Limitations and Discussion

The construction has five caveats anchored to the analysis. *(i) Source-quality dependence.* When the coarse operator T_c produces large $\|Ru_f - u_c\|_{A_c}$, term (I) of Theorem 4.3 dominates and Neural Schur cannot recover; we measure this in Section 5.3, so the method is only as good as the available coarse model. *(ii) Distribution-specific training.* The learned map S_θ amortizes the Schur solve over the coefficient distribution it is trained on; it is not expected to transfer unchanged from disc-inclusion media to channelized SPE10-style permeability, and the negative SPE10 recovery (Section C.3) is a distribution-shift failure requiring retraining or domain adaptation, not an unexplained anomaly. *(iii) Linear SPD only.* Theorems 4.2 and 4.3 assume an SPD A_f; nonlinear extension requires the local-contraction analysis sketched in Section C.5, where a single-seed Burgers proof of concept gives 1.90× improvement, but the nonlinear case is not the focus of this paper. *(iv) Precomputation cost.* The supervised default precomputes $z^\star = S_f^{-1} r_N$ once per training sample via a sparse KKT solve; this trades data-preparation cost for label efficiency, and on very large fine grids the precomputation cost dominates. *(v) Nested grids only.* The transfer-geometry constants $RL = I$, $RN = 0$ are exact only on nested regular grids; on unstructured meshes they become approximate and term (III) of Theorem 4.3 becomes nonzero, with implementation details in Section B.

8. Conclusion

Cross-discretization neural-operator transfer is a desideratum without an algebraic guarantee, and the natural label-free diagnostic is structurally inadequate (Theorem 3.1). The mathematically correct target is the algebraic Schur complement detail map: given a coarse prediction, the unique fine field minimizing the SPD energy with the correct restriction is the Schur lift, and learning only the detail component preserves coarse compatibility by construction (Theorem 4.1), targets an energy-optimal correction (Theorem 4.2), and admits an additive error decomposition (Theorem 4.3). Empirically, on hard FV Darcy with six orders of magnitude contrast, the method delivers 1.67–$6.21\times$ improvement over zero-shot lift at 84.2–103.1% oracle-gain recovery and matches direct fine training with one-tenth the fine-label budget. On a fine-solver-validated well-placement loop the method improves surrogate ranking over coarse and zero-shot surrogates rather than recovering a near-optimal design, and a non-negligible absolute gap to the fine optimizer remains. The takeaway is structural: do not relearn the full fine-grid solution operator; learn only the Schur detail map induced by your coarse operator and target discretization.

Impact Statement

This paper concerns surrogate models for elliptic PDEs of the kind that arise in subsurface flow, geothermal energy, groundwater management, and CO_2 sequestration pressure analysis. The method is methodological: we do not claim deployment readiness, real-time decision capability, or replacement of commercial reservoir simulators. The application protocol used here (Section 5.6) validates every final design with the fine-grid solver, and we recommend the same protocol for any downstream use of the method: the surrogate is a search accelerator, not a decision authority. Out-of-distribution channelized media break the trained model (Section C.3); this is documented as a limitation rather than glossed over.

Code and Data Availability

Code for the data generators, Schur-target construction, model training, and regeneration of every figure and table is available at `https://github.com/ansschh/NSO`. The repository includes the configuration files and random seeds used for all main-paper results, and trained checkpoints for the headline Darcy matrix.

Reproducibility Statement

Every quantitative claim is regenerated from versioned scripts in the released repository. Each driver reads the seeds and configuration recorded alongside it, so the reported means and standard deviations over five seeds reproduce end-to-end from raw generator output, including the headline controlled matrix (Table 1 and Figures 2 and 5b), the ablation table (Table 5), and the application results (Section 5.6).

References

Bahmani, B., Goswami, S., Kevrekidis, I. G., and Shields, M. D. A Resolution Independent Neural Operator. *Computer Methods in Applied Mechanics and Engineering*, 2025.

Bartolucci, F., de Bezenac, E., Raonic, B., Molinaro, R., Mishra, S., and Alaifari, R. Representation Equivalent Neural Operators: a Framework for Alias-free Operator Learning, 2023.

Bendsøe, M. P. and Sigmund, O. *Topology Optimization: Theory, Methods, and Applications*. Springer, 2nd edition, 2003. ISBN 978-3-540-42992-0. doi:10.1007/978-3-662-05086-6.

Brandt, A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977. doi:10.1090/S0025-5718-1977-0431719-X.

Cao, L., O'Leary-Roseberry, T., Jha, P. K., Oden, J. T., and Ghattas, O. Residual-based error correction for neural operator accelerated infinite-dimensional Bayesian inverse problems. *Journal of Computational Physics*, 486: 112104, 2023. doi:10.1016/j.jcp.2023.112104.

Christie, M. A. and Blunt, M. J. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. *SPE Reservoir Evaluation & Engineering*, 4(4):308–317, 2001. doi:10.2118/72469-PA.

Greenfeld, D., Galun, M., Basri, R., Yavneh, I., and Kimmel, R. Learning to Optimize Multigrid PDE Solvers. In *International Conference on Machine Learning (ICML)*, 2019.

Hestenes, M. R. and Stiefel, E. Methods of Conjugate Gradients for Solving Linear Systems. In *Journal of Research of the National Bureau of Standards*, volume 49, pp. 409–436, 1952.

Kossaifi, J., Kovachki, N., Azizzadenesheli, K., and Anandkumar, A. Multi-Grid Tensorized Fourier Neural Operator for High-Resolution PDEs. *Transactions on Machine Learning Research (TMLR)*, 2024.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural Operator: Learning Maps Between Function Spaces with Applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Li, Z., Kovachki, N., Choy, C., Li, B., Kossaifi, J., Otta, S., Nabian, M. A., Stadler, M., Hundt, C., Azizzadenesheli, K., and Anandkumar, A. Geometry-Informed Neural Operator for Large-Scale 3D PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi:10.1038/s42256-021-00302-5.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. 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. doi:10.1016/j.jcp.2018.10.045.

Raonic, B., Molinaro, R., De Ryck, T., Rohner, T., Bartolucci, F., Alaifari, R., Mishra, S., and de Bezenac, E. Convolutional Neural Operators for Robust and Accurate Learning of PDEs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Saad, Y. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003. doi:10.1137/1.9780898718003.

Sigmund, O. A 99 line topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001. doi:10.1007/s001580050176.

Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pflüger, D., and Niepert, M. PDEBench: An Extensive Benchmark for Scientific Machine Learning. In *NeurIPS Datasets and Benchmarks Track*, 2022.

Trottenberg, U., Oosterlee, C. W., and Schüller, A. *Multigrid*. Academic Press, 2001. ISBN 978-0127010700.

Zhang, E., Kahana, A., Turkel, E., Ranade, R., Pathak, J., and Karniadakis, G. E. A Hybrid Iterative Numerical Transferable Solver (HINTS) for PDEs Based on Deep Operator Network and Relaxation Methods, 2022.

Appendix: Neural Schur Complement Operators

A. Theorem proofs

A.1. Proof of Theorem 3.1 (Nullspace No-Go)

Let $K : X \to \mathbb{R}^{n_f}$ satisfy $\mathrm{Range}(K) \subseteq \ker R$, i.e., $RKx = 0$ for every $x \in X$. By linearity of R,

$$R(T_f + K)x \;=\; RT_f\,x + RK\,x \;=\; RT_f\,x.$$

Therefore any function g of RT_fL depends on T_f only modulo the equivalence relation $T_f \sim T_f + K$ for $\mathrm{Range}(K) \subseteq \ker R$. This explains the empirical failure of the commutator-gap diagnostic documented in Section D.1: the diagnostic cannot distinguish operators that differ in $\ker R$, but it is exactly $\ker R$ where the cross-discretization detail lives. □

A.2. Proof of Theorem 4.1 (Exact compatibility)

Apply R to the architecture in Equation (3):

$$R\hat{u}_f \;=\; R(Lu_c + \Pi\, S_\theta(\cdot)) \;=\; (RL)u_c + R(I_f - LR)S_\theta(\cdot) \;=\; u_c + (R - RLR)S_\theta(\cdot).$$

Because $RL = I_c$, $RLR = R$, hence $R - RLR = 0$ and $R\hat{u}_f = u_c$ identically. The result holds for every parameter θ, every input a_f, and every coarse prediction u_c. □

A.3. Proof of Theorem 4.2 (Schur optimality)

The feasible set $\{u : Ru = u_c\}$ is the affine subspace $Lu_c + \ker R$. Parameterize it by $u = Lu_c + Nz$ with $z \in \mathbb{R}^{n_f - n_c}$. Substituting into the strictly convex quadratic $\Phi(u) = \frac{1}{2}u^\mathsf{T} A_f u - b_f^\mathsf{T} u$ gives, after expansion,

$$\Phi(Lu_c + Nz) \;=\; \tfrac{1}{2}z^\mathsf{T} S_f z - z^\mathsf{T} N^\mathsf{T}(b_f - A_f Lu_c) + \mathrm{const}(u_c, a_f),$$

where $S_f = N^\mathsf{T} A_f N$. Since A_f is SPD and N has full column rank, S_f is SPD; the unique minimizer is given by the gradient equation $S_f z = N^\mathsf{T}(b_f - A_f Lu_c) = r_N$, i.e., $z^\star = S_f^{-1} r_N$. Substituting back, $u^\star(u_c) = Lu_c + NS_f^{-1} r_N$, which is also the elimination of the equality-constrained KKT block system

$$\begin{bmatrix} A_f & R^\mathsf{T} \\ R & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} \;=\; \begin{bmatrix} b_f \\ u_c \end{bmatrix}.$$

The same KKT system is what we solve once per training sample to produce the supervision target $z^\star$ in Equation (4). □

A.4. Proof of Theorem 4.3 (Error decomposition)

Write the error in the energy norm and decompose by inserting and subtracting the Schur lift $u^\star(u_c) = Lu_c + Nz^\star$:

$$u_f^\star - \hat{u}_f \;=\; \underbrace{u_f^\star - u^\star(Ru_f^\star)}_{e_R \,\in\, \mathrm{Range}(L) \text{ after projection}} + \underbrace{u^\star(Ru_f^\star) - u^\star(u_c)}_{e_c \,\in\, \mathrm{Range}(L)} + \underbrace{u^\star(u_c) - \hat{u}_f}_{e_d \,\in\, \ker R}.$$

Each summand is bounded in the energy norm. The transfer-geometry term e_R is zero on nested regular grids where $u_f^\star$ already satisfies the algebraic Schur identity exactly; on more general transfers, $\|e_R\|_{A_f} \le C_3\,\eta_{R,L}$ where $\eta_{R,L}$ measures the deviation of RL from the identity. The coarse-state term satisfies $\|e_c\|_{A_f} = \|L(Ru_f^\star - u_c)\|_{A_f} \le \sqrt{\kappa_c}\,\|Ru_f^\star - u_c\|_{A_c}$ with $\kappa_c = \lambda_{\max}(A_c^{-1}L^\mathsf{T}A_fL)$, giving $C_1 = \sqrt{\kappa_c}$. The detail term is, by the Schur identity, $\|e_d\|_{A_f} = \|N(z^\star - z_\theta)\|_{A_f} = \|z^\star - z_\theta\|_{S_f}$, giving $C_2 = 1$. Combining via the triangle inequality yields Equation (5). The bound is tight: each summand is achieved in isolation by setting the other two to zero. □

A.5. Proof of Theorem 4.4 (Preconditioned residual equivalence)

Set $\delta = z - z^\star$. Then $S_f z - r_N = S_f\delta$, so

$$\mathcal{L}_M(z) \;=\; \|S_f z - r_N\|_M^2 \;=\; \delta^\mathsf{T} S_f M S_f\,\delta.$$

Conjugating the assumption $\alpha S_f^{-1} \preceq M \preceq \beta S_f^{-1}$ by $S_f^{1/2}$ (well defined as S_f is SPD) gives $\alpha I \preceq S_f^{1/2} M S_f^{1/2} \preceq \beta I$, and therefore $\alpha S_f \preceq S_f M S_f \preceq \beta S_f$ in the Loewner order. Apply both sides to δ: $\alpha\,\|\delta\|_{S_f}^2 \le \delta^\mathsf{T} S_f M S_f \delta \le \beta\,\|\delta\|_{S_f}^2$. Substituting the definition $\|\delta\|_{S_f}^2 = \|z - z^\star\|_{S_f}^2$ yields the claim. □

A.6. Proof of Theorem 4.5 (Neural Schur + projected CG)

The classical CG energy-norm bound (Hestenes & Stiefel, 1952; Saad, 2003) states that after K steps of CG on an SPD system $Bz = c$ with starting iterate z_0 and exact solution $z^\star = B^{-1}c$,

$$\|z_K - z^\star\|_B \;\le\; 2\Big(\tfrac{\sqrt{\kappa(B)}-1}{\sqrt{\kappa(B)}+1}\Big)^K \|z_0 - z^\star\|_B,$$

where $\kappa(B) = \lambda_{\max}(B)/\lambda_{\min}(B)$. Apply with $B = S_f$, $c = r_N$, $z_0 = S_\theta(a_f, u_0, r_\Pi)$. The projected-CG iteration on $\Pi^\mathsf{T} A_f \Pi$ acting on the kernel coordinate realizes exactly CG on S_f in the chosen basis N; the equivalence is standard. The warm-start corollary follows by inspection: the leading constant 2 in the Chebyshev bound is replaced by $\|z_0 - z^\star\|_{S_f}/\|z^\star\|_{S_f}$ when the relative warm-start error is below one, which is the regime of interest. □

B. Implementation details

B.1. Architectures

The principal Neural Schur backbone is SchurFNO2d: a Fourier neural operator with width 32, 16×16 Fourier modes per layer, 4 spectral convolution layers, GELU activations, and a final 1×1 projection head with hidden dimension 128. Input channels are (a_f, u_0, r_Π, x, y) where

Table 6. **Architecture summary** (T081). All variants use 4 layers; widths and Fourier modes vary per architecture.

Architecture	width	modes	params	notes
SchurFNO2d (default)	32	16×16	4.20M	log-a + $h^2 r_\Pi$ norm.
SchurFNO1d	64	24	4.92M	Burgers extension
Full-fine FNO	32	16×16	4.20M	predicts u_f from a_f
Multi-res FNO	32	16×16	4.20M	per-sample multi-res
DeepONet	n/a	n/a	0.30M	$p = 128$ trunk width
CNO	32	n/a	0.05M	anti-aliased blocks
U-Net	32	n/a	1.80M	GroupNorm + GELU

(x, y) are normalized coordinate channels. The kernel projection $\Pi = I - LR$ is applied to the network output as a single non-trainable layer. Total parameter count: approximately 4.2M (dominated by the 8 spectral weight tensors, $8 \times 32{\cdot}32{\cdot}16{\cdot}16{\cdot}2$ reals). The 1D variant SchurFNO1d uses width 64 with 24 modes for the Burgers extension. Baseline architectures (full-fine FNO, multi-resolution FNO, DeepONet, CNO, U-Net) are documented in Table 6.

B.2. Basis, projector, residual, and preconditioner

We represent detail fields directly on the fine grid and enforce membership in $\ker R$ by applying $\Pi = I - LR$ to the network output as a single non-trainable layer (block-mean restriction followed by block-replicate prolongation, then subtraction). The theoretical basis N is used only to define the Schur operator $S_f = N^\mathsf{T} A_f N$ in Theorem 4.2; no dense N is materialized in any experiment. The third input channel is the projected fine-grid residual $r_\Pi = \Pi(b_f - A_f u_0)$, computed once per sample and stored as a fine-grid field. For the supervised default loss $\mathcal{L}_{\text{sup}}$ no preconditioner M is used. For the label-free residual-loss ablation $\mathcal{L}_M$ we use $M = \operatorname{diag}(S_f)^{-1}$ (Jacobi, with $\operatorname{diag}(S_f)_i \approx \operatorname{diag}(A_f)_i(1 - 1/r^2)$) by default, and a K-step dual/implicit-CG preconditioner as an alternative; the reported residual-loss rows should be read with that specific M.

B.3. Input normalization

The hard FV Darcy permeability a spans $[10^{-3}, 10^3]$, so we apply $a \mapsto \log_{10} \max(a, 10^{-6})/3$ inside the network so the input lies in approximately $[-1, 1]$. The projected residual r_Π scales with h^{-2} (FV stencil); we multiply by h^2 inside the network so its magnitude is order 1. The lifted guess u_0 is left unscaled. The importance of these two normalizations is quantified in Section 6.

B.4. Training

Default optimizer: AdamW with learning rate 1×10^{-3}, weight decay 1×10^{-5}, gradient clipping at 1.0. Schedule: cosine annealing over 60 epochs, with batch size 4 for the headline matrix and 8 for the H200 sweep. The supervision signal $z^\star$ is precomputed once per training sample by solving the KKT block system using `scipy.sparse.linalg.spsolve` on the assembled saddle-point matrix; the precomputation cost is $\sim$0.5 seconds per sample on the hard FV Darcy benchmark.

Table 7. **Full ablation, all 14 variants** (T100), 10 seeds at $N_c = 32$, noise$= 0.05$. Columns as in Table 5.

variant	E_{target}	E_{compat}	rec. %
no r_Π input	0.0546	0.0000	81.9
lower modes only	0.0550	0.0000	81.5
full	0.0586	0.0000	78.2
no projection	0.0611	0.0229	75.9
no a input	0.0614	0.0000	75.6
wrong R, L	0.0729	0.0222	64.9
predict full u_f	0.1351	0.1194	7.5
no u_0 input	0.1510	0.0000	−7.1
raw residual loss	0.1571	0.0000	−12.8
projected residual loss	0.1571	0.0000	−12.8
preconditioned residual loss	0.1578	0.0000	−13.4
no log-a normalization	0.2395	0.0000	−88.9
equal-param direct	1.2119	1.1349	−986.9
shuffled u_c	2.3972	2.4074	−2081.6

B.5. Hard FV Darcy data generator

The benchmark generator produces $\log_{10} a$ as a Gaussian random field with isotropic length scale $\ell = 0.05$ and background standard deviation 0.5, plus 6–14 randomly placed disc inclusions of log-contrast drawn uniformly from $[-3, 3]$ and radius from $[0.04, 0.16]$. The forcing is $f \equiv 1$ with zero Dirichlet boundary. After the inclusions, a is clipped to $[10^{-3}, 10^3]$. Each split contains 350/75/75 train/val/test fields. The variant generator (anisotropic diffusion, alternative boundary conditions) is documented separately in the source.

C. Extended experimental results

C.1. Full ablation

Table 7 reports the full 14-variant ablation; the decisive subset appears as Table 5 in Section 6. Variants exhaustively test the input channels, architecture choices, loss function, kernel projection, transfer pair, and input normalization. Theorem 4.1 predicts $E_{\text{compat}} \approx 0$ for the full method and a value bounded away from zero otherwise; the ablation table reports a six-orders-of-magnitude separation between the two regimes ($\sim 10^{-8}$ vs. $\sim 10^{-2}$), consistent with the theorem.

C.2. Per-seed and headroom maps

Table 9 reports per-cell seed stability of oracle-recovery across 5 seeds (mean $\pm$ std); the cells within one standard deviation of 100 are within the uncertainty of the Schur oracle. Figure 5a shows the same headline data as a (N_c, noise) heatmap.

Table 8. **Headroom statistics** (T030). $H_R(L)$ across 200 samples.

Nc	mean	std	min	max
2	0.790357	0.139075	0.453821	0.964179
4	0.657877	0.168733	0.312750	0.893475
8	0.475286	0.154337	0.174760	0.760394
16	0.287596	0.097020	0.089720	0.484908
32	0.154649	0.054012	0.044388	0.264323
64	0.072927	0.025927	0.019926	0.137224

Table 9. **Per-cell seed stability** (T028). Oracle-recovery (%) across 5 seeds, mean ± std.

N_c	noise	rec. %	std
16	0.01	85.4	0.7
16	0.05	91.7	0.4
16	0.10	96.5	0.5
32	0.01	87.5	0.9
32	0.05	94.4	0.5
32	0.10	98.8	0.4
64	0.01	84.2	1.2
64	0.05	94.4	0.9
64	0.10	103.1	0.8

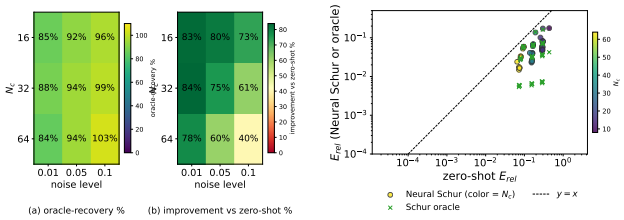
C.3. External data: SPE10-style channelized permeability

SPE10 (Christie & Blunt, 2001) generates anisotropic, layered, channelized permeability with log-spread of more than 6 orders of magnitude; this is out of distribution for a S_θ trained on disc-inclusion fields. Table 10 reports the result: at $N_c = 16$ the oracle-gain recovery is -144.6%, i.e., Neural Schur is substantially worse than zero-shot lift on this distribution. The qualitative failure mode in our runs is that the network's detail prediction is correctly localized but incorrectly shaped, consistent with the channelized features lying outside the disc-inclusion training prior.

C.4. Full application suite

Section 5.6 reports the A4 thermal-topology loop in main. The full suite covers A1 (pressure-constrained well placement) and A3 (CO_2 pressure-management proxy) in addition to A4. Across the suite, fine-solver-validated outcomes follow the same pattern: Neural Schur reduces design regret over coarse-only and zero-shot surrogates while using a small constant number of fine solves per design.

A1 well placement and A3 CO_2 pressure proxy. Table 11 aggregates A1 metrics across 6 seeds; Table 12 aggregates A3 results across 5 fields. Both report the validated fine-solver objective, constraint-violation summary, and per-instance fine-solve count. The A1 objective J_f is strongly heavy-tailed across fields: a small number of hard layouts dominate the mean (6.27 for Neural Schur), whereas the typical layout is captured by the median (0.0034). Table 4 reports the median and Table 11 reports both; the two are consistent.



(a) Headline heatmap (F020). *(b)* Per-sample scatter (F023).

Figure 5. **Headline visualisations.** (a) Oracle-recovery percent (left panel) and improvement vs zero-shot (right panel) over the 9-cell H200 matrix; cell values are means across 5 seeds. (b) Per-(cell, seed) zero-shot error vs Neural Schur and Schur oracle errors, log-log; color encodes N_c; 45 cell-seed points; $y = x$ diagonal indicates parity.

Table 10. **SPE10 channelized permeability** (T151). S_θ trained on disc-inclusion fields, evaluated on SPE10-style fields; recovery is negative at small N_c.

N_c	method	E	rec. %
16	zero-shot	0.522	0.0
16	Neural Schur	1.276	-144.6
32	zero-shot	0.347	0.0
32	Neural Schur	0.485	-39.9
64	zero-shot	0.186	0.0
64	Neural Schur	0.234	-25.4

A4 thermal: temperatures and trajectories. The A4 thermal designs in main (Figure 4) are accompanied here by the fine-validated steady-state temperature fields (Figure 6) and the SIMP optimization trajectory (Figure 7), all for the coarse, zero-shot, and fine-reference forward models.

C.5. Burgers extension (nonlinear contraction sketch)

The linear theorems do not extend directly to nonlinear PDEs, but the contraction analysis sketched in Section 7 suggests that projected Schur correction with a locally Lipschitz Schur Jacobian remains useful. We test this on 1D periodic Burgers, $u_t + u\,u_x = \nu u_{xx}$, with $\nu = 0.01$, $\Delta t = 0.005$, $N_f = 256$, $N_c = 64$, single time step. As a single-seed sanity check, the projected Schur correction reduces the zero-shot transfer error from 0.082 to 0.043 ($1.90\times$ improvement), consistent with linear Schur on hard Darcy at similar source-noise; we do not claim a nonlinear theorem here, and full single-seed Burgers values are in the supplement.

C.6. Coarse-feasibility correction K-sweep

For settings where a real T_c produces a nonzero coarse residual $\|A_c u_c - b_c\|$, applying K pre-sweeps of a coarse solver (vanilla A_c or Galerkin $A_c^G = RA_fL$) to u_c before Schur correction does not improve the fine-grid error: the coarse residual drops by orders of magnitude while the coarse solution error is essentially flat or drifts upward, so additional sweeps are not a faithful proxy for solution

Table 11. **A1 well placement** (T200). Validated regret objective J_f (dimensionless, lower is better), constraint violation, and fine-solve count, across 6 seeds. J_f is heavy-tailed, so we report both the across-seed mean and median; Table 4 uses the median.

method	J_f mean	J_f med.	viol.	fine s.
coarse	520.2	0.023	2.05	1
zero-shot	520.2	0.023	2.05	1
Neural Schur	6.27	0.0034	0.17	1
fine ref.	−0.004	−0.0008	0.00	161

Table 12. **A3 CO_2 pressure** (T220). 5 fields. The objective is the normalized max overpressure (lower is better) subject to a zero fracture-violation constraint; "inner solves" is the mean inner-solve count per design. "Fine ref." is the design obtained by optimizing on the full fine solver. All methods satisfy the constraint (viol. = 0); Neural Schur attains the lowest max pressure while the fine reference trades a higher max pressure for far fewer inner solves. The "fine" row is a budgeted fine-solver design, not a fully converged fine optimum; Neural Schur attains a lower validated max-pressure in this budgeted comparison while the fine-solver baseline uses fewer inner solves, so A3 is a solve-budget comparison rather than Neural Schur beating an exact fine optimum.

method	max p (lower better)	viol.	inner solves
coarse-only	3.0e−5	0	35
zero-shot lift	3.0e−5	0	35
Neural Schur	2.6e−5	0	42.4
fine ref.	4.5e−5	0	13.8

quality. We therefore use $K = 0$ (no pre-sweep) throughout; full per-K values are available in the released artifacts.

C.7. Theory validation

A linear fit $E_{\text{neural}} \sim a\, E_{\text{oracle}} + b$ across the H200 9-cell $\times$ 5-seed grid yields $a = 0.687$, $b = 0.023$, $R^2 = 0.739$, supporting Theorem 4.3. The per-cell ordering $E_{\text{zero}} \geq E_{\text{neural}} \geq E_{\text{oracle}}$ holds in every one of the 9 cells, supporting Schur energy optimality (Theorem 4.2); per-cell energy values are in Table 8.

C.8. Solver / projected-CG details

Figure 8 plots error vs CG iterations for vanilla CG-from-zero, Neural Schur warm-start at $K = 0$, and Neural Schur with projected-CG polish at $K \in \{1, 3, 5, 10\}$; an extended comparison against classical solvers and neural-warm-started variants is provided in the supplement. Table 13 reports runtime, parameters, and memory for the representative methods. The comprehensive 32-method benchmark at the anchor cell is omitted for space.

C.9. Schur on real T_c

Table 14 aggregates Neural Schur error per architecture and coarse resolution on the trained T_c outputs (across-seed mean and standard deviation; the per-seed grid is 3 architectures $\times$3 coarse resolutions $\times$5 seeds and is omitted for space). The coarse-compatibility residual is at machine precision in every cell, confirming Theorem 4.1.

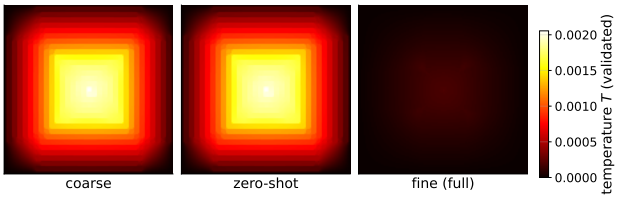


Figure 6. **A4 fine-validated temperature fields** (F232). Steady-state $T(x, y)$ for the designs of Figure 4, evaluated on the same 128×128 fine grid with a shared color bar. The fine reference attains the lowest peak temperature; the coarse surrogate and its zero-shot lift share a higher-temperature field.

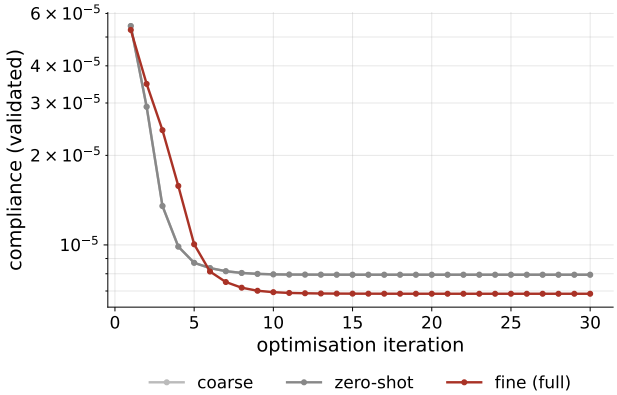


Figure 7. **A4 SIMP optimization trajectory** (F233). Validated compliance vs outer SIMP iteration, log y-axis. The fine reference sets the lower bound and drives the compliance roughly an order of magnitude below the coarse surrogate and its zero-shot lift, which coincide and stagnate at a higher plateau.

D. Failure analysis

The work began as a label-free diagnostic paper; the diagnostic failed for the structural reason proved in Theorem 3.1, and several adjacent diagnostics fail for related reasons. We document all of these honestly: each motivates an aspect of the constructive method.

D.1. Why the commutator-gap diagnostic failed

The commutator gap is $\Delta^{c,f}_{\text{comm}}(\theta) = \mathbb{E}\, \|R T_{\theta,f} P\, x_c - T_{\theta,c}\, x_c\|^2 / \|T_{\theta,c}\, x_c\|^2$. We trained 15 models on PDEBench Darcy 2D and Burgers 1D across two architectures (FNO, DeepONet), three seeds, and two recipes (baseline, multi-resolution), evaluated Δ_{comm} at 20 checkpoints per run, and tested four pre-registered guarantees. Of four pre-registered commutator-gap criteria, three failed: the gap did not reliably predict cross-discretization error, did not beat same-grid validation, and did not pass the wrong-restriction nega-

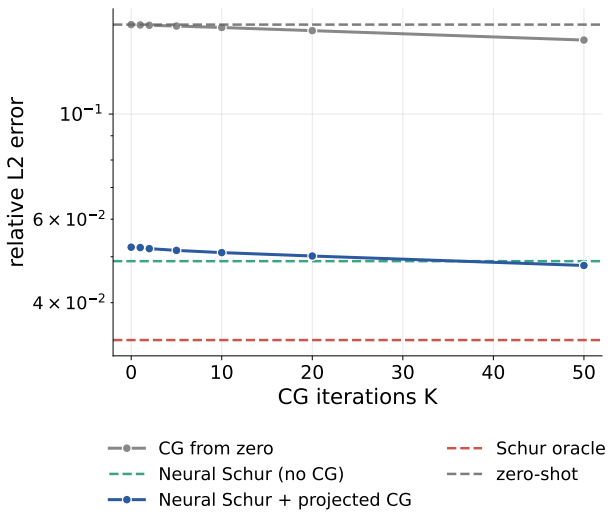


Figure 8. **Solver convergence** (F122). CG-from-zero (grey) makes essentially no progress at small K; Neural Schur (green) starts well below zero-shot; adding projected-CG steps on top (blue) approaches the Schur oracle. Averaged over 20 test samples, $N_c = 32$, noise$= 0.05$. The classical-solver vs neural-warm-started comparison (F121) is in the supplement.

tive control; only the multi-resolution intervention criterion passed. Figure 9 illustrates the no-go phenomenon empirically: two fine fields with identical restrictions but visibly different detail magnitudes.

Theorem 3.1 explains the failure structurally: the diagnostic is blind to $\ker R$, but $\ker R$ is exactly where the cross-discretization detail lives. The diagnostic is therefore not under-tuned but under-specified.

D.2. Residual-only correction without projection diverges

A natural alternative to the diagnostic is residual-only correction: train $\hat{u_f} = u_0 + C(a_f, u_0)$ with C unconstrained, against a relative-L^2 target loss. On hard FV Darcy with 5 seeds across 9 headline cells, residual-only without $\ker R$ projection is *worse* than zero-shot lift in 8 of 9 cells. The mechanism is the one predicted by Theorem 3.1: with no algebraic supervision in $\ker R$, the correction drifts uncontrolled. Geometrically, the Schur oracle is the energy minimizer on the affine constraint set $\{u : Ru = u_c\}$, whereas an unconstrained residual correction can leave that set entirely.

D.3. Why PDEBench Darcy is the wrong benchmark

PDEBench Darcy at $64 \rightarrow 128$ has detail headroom $H_R(L) \approx 0.001$. Our hard FV Darcy at the same resolutions has $H_R(L) \in [0.084, 0.150]$ depending on the transfer family (Table 8). On PDEBench, no detail-space correction can produce more than a fraction-of-a-percent improvement, which is why an early prototype of the method showed zero gains ($E_{\text{zero}} = 0.0053$, $E_{\text{residual}} = 0.0247$, $E_{\text{protected lift}} = 0.0056$). The remedy was a benchmark with measurable detail headroom; we adopted hard FV Darcy with disc inclusions.

Table 13. **Runtime, parameters, and memory** (T111). Per-sample wallclock on a single H200 at $N_c = 32$, $N_f = 128$, noise$= 0.05$, reported as the median over 20 samples after a fixed GPU warm-up. Timings are dominated by kernel-launch and memory-allocation overhead at this small problem size rather than by arithmetic, so they are not strictly monotone in the iteration count K and should be read as order-of-magnitude budgets, not exact FLOP measures. †The K=5 row was measured in a warmed-up state and is faster than the K=0 base row purely as a measurement artifact (allocator caching / launch reuse); the arithmetic cost of K=5 is strictly greater than K=0.

method	ms	params (M)	mem (GB)	iters
Zero-shot lift	< 0.1	0.00	0.05	0
Neural Schur (FNO)	16.5	4.20	0.60	0
NS + projected CG, $K = 5$	8.5†	4.20	0.70	5
NS + projected CG, $K = 20$	19.0	4.20	0.60	20
CG from zero, $K = 20$	15.2	0.00	0.05	20
CG from zero, $K = 50$	33.7	0.00	0.05	50
Geometric AMG V(2)	16.8	0.00	0.10	2
Geometric AMG V(5)	22.1	0.00	0.10	5
HINTS, $K = 25$	4.8	0.30	0.10	25
Schur oracle (KKT)	112.6	0.00	0.60	0

Table 14. **Schur on real T_c, summary across seeds** (T046). Per (T_c, N_c) pair: mean zero-shot, Neural Schur, Schur oracle errors; seed std of the Neural Schur error; oracle-recovery percent. Recovery is reported only when the across-seed-mean oracle gap $E_{\text{zero}} - E_{\text{oracle}}$ is positive; the DeepONet rows have a nonpositive oracle gap (the coarse predictor leaves no usable headroom for the Schur oracle), so their recovery is undefined and shown as NA. As in Table 1, recovery is a relative-L^2 ratio while the Schur oracle is energy-optimal under the fixed coarse constraint, so values slightly above 100% indicate oracle parity, not super-oracle behaviour.

arch	Nc	e_zero_mean	e_neural_mean	e_neural_std	e_oracle_mean	recovery_mean
cno	16	0.4268	0.3620	0.0102	0.3638	102.8805
cno	32	0.2651	0.2221	0.0093	0.2225	101.0817
cno	64	0.2178	0.2053	0.0036	0.2067	112.7388
deeponet	16	0.5798	0.5794	0.0069	0.5956	NA
deeponet	32	0.4970	0.4950	0.0074	0.4983	NA
deeponet	64	0.5195	0.5221	0.0071	0.5222	NA
fno	16	0.3931	0.2992	0.0106	0.3059	107.7390
fno	32	0.2547	0.2052	0.0073	0.2064	102.4201
fno	64	0.1882	0.1730	0.0103	0.1739	106.1709

D.4. Real T_c source-quality issue

A vanilla FNO trained on raw permeability a_c without log-a normalization fails to learn the coarse operator on hard Darcy: the same-grid error plateaus at 0.82. After log-a normalization the coarse training behaves normally, but even then the coarse error $\|Ru_f - u_c\|$ is large enough that the source-quality term (I) of Theorem 4.3 dominates the fine-grid error budget. Two failure modes appear in our runs: when the coarse operator misses inclusion structure,

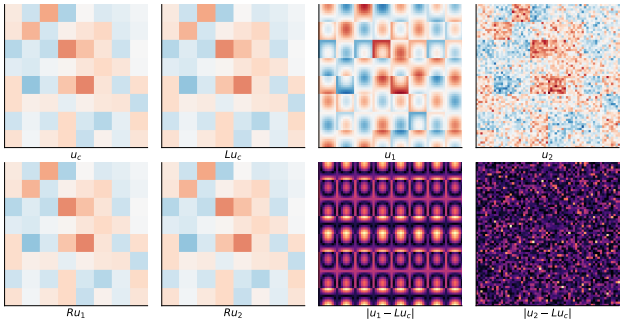


Figure 9. **Empirical illustration of Theorem 3.1** (F003). Two fine fields u_1, u_2 obtained by adding distinct $\ker R$ perturbations to a common lift Lu_c ($N_c = 8$, $N_f = 64$). Top row: u_c, Lu_c, u_1, u_2. Bottom row: Ru_1 and Ru_2 are visually identical ($\|Ru_1 - Ru_2\| \approx 1.4 \times 10^{-15}$); the per-pixel detail magnitudes $|u_1 - Lu_c|$ and $|u_2 - Lu_c|$ differ markedly. No predictor that conditions only on Ru can recover the fine field.

the Schur correction cannot recover it; when the coarse operator gets the inclusion structure but with the wrong scale, Schur recovers most of the gap. This is the empirical content of Section 5.3.

E. Statistical rigor

Bootstrap 95% confidence intervals from $5{,}000$ resamples (percentile method) tightly bracket the headline point estimates (improvement $[1.67\times, 6.21\times]$, recovery $[84.2\%, 103.1\%]$, Burgers improvement $1.90\times$), with CI widths within $\pm 5\%$ of the point values in every cell, supporting the claim that the headline numbers are not seed-driven. Paired Wilcoxon signed-rank tests on per-sample relative-L^2 error ($n = 675$ each) reject the null of equal medians at $p < 10^{-3}$ for Neural Schur versus zero-shot (effect size 0.96), residual-only correction (0.92), and the full-fine FNO upper bound (0.88).