LEARNING WHAT MULTIGRID LEAVES BEHIND: OBJECTIVE ALIGNMENT AND ITS LIMITS

Anonymous authors
Paper under double-blind review

ABSTRACT

A learned coarse space adds a low-rank correction to a multigrid preconditioner, but which space should the network predict? We separate three requirements. *Target*: the training target should complement the deployed base preconditioner, not only the operator's low-energy modes. *Prediction*: a network must predict that target on new operators. *Cost*: the correction must beat strong classical solvers in total time. For a standard two-level balancing correction, a known result says the condition-number-optimal space consists of the directions the base preconditioner reduces most slowly. We train an FNO on a finite solver target that approximates these directions by repeatedly applying the preconditioner's error propagator to random vectors. With architecture, rank, training operators, and deployment fixed, this target raises the mean iteration reduction on 24 held-out 2D operators from 17.6% (operator-only Rayleigh training) to 31.7%, beating even the Rayleigh objective's exact optimum (18.8%), and targets crossed between two preconditioners lose on every operator. The target requirement therefore holds in this 2D family, and a useful target also exists in a 3D obstacle family. Prediction holds only inside the training band, where it is slightly better than reusing a stored target; it fails on disjoint angles, on lognormal coefficients, and in 3D. Cost favors classical solvers: the corrected preconditioner needs about as many iterations as two V-cycles and more flops, and BoomerAMG is faster in total time.

1 INTRODUCTION

Many simulations solve the same kind of linear system thousands of times: implicit time steps on a fixed geometry, parameter sweeps, design loops. Multigrid is a standard solver for such systems, and what limits it is a small set of error components that its smoother removes slowly. A coarse space represents those components. Choosing it is central to multigrid and domain decomposition, and it is an appealing learning problem: a network can inspect the operator and propose these few nonlocal directions before the solve begins. Recent learned solvers often report large gains over a default classical configuration (Greenfeld et al., 2019; Luo et al., 2024; Fink et al., 2026). The apparent gain can depend strongly on the classical configuration used as the baseline: a different strength measure, smoother, or coarse-space budget can change the conclusion without changing the learned model. If the value of a learned space depends on the preconditioner it corrects, so should its training target. This paper asks what a learned coarse space must achieve to be worth using, and separates three requirements that are easy to conflate.

1. *Target:* does the training target complement the deployed base preconditioner?
2. *Prediction:* can a network predict that target on new operators?
3. *Cost:* does the resulting correction beat strong classical alternatives in total time?

The target requirement has a precise answer for a standard two-level balancing correction: the best rank-m space consists of the directions that the base preconditioner reduces most slowly. This is known in AMG and deflation theory (Xu & Zikatanov, 2017; Brannick et al., 2017; Tang et al., 2009); we restate it for our correction (Lemma 1) because it names the target a network should learn. The operator-only Rayleigh objective, a standard spectral-learning loss (Pfau et al., 2018; Deng et al., 2022) and the loss of the Rayleigh baseline we start from, instead selects the lowest-energy directions of the normalized operator, which need not be the directions the preconditioner

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

handles poorly. We call conditioning the target on the base preconditioner *solver alignment*, the objective alignment of our title.

We test the three requirements in a controlled 2D family with non-grid-aligned anisotropy, a known weakness of classical strength heuristics, and in a 3D family whose low-conductivity barriers stress geometric multigrid. We then challenge the conclusions with public sandstone, trabecular-bone, and SPE10 data and a HYPRE Example 7 benchmark (Jackson et al., 2021; Kuczynski, 2024; Christie & Blunt, 2001). Appendix Table 5 summarizes where each requirement holds. The target requirement holds: in 2D, training against a finite solver target beats Rayleigh training and even the exact Rayleigh space, and the useful target moves with the preconditioner. Prediction is the bottleneck: it works inside the training band but not beyond it, and not in 3D. Cost favors classical solvers once the baseline is strong.

Specifically, we contribute:

1. a restatement of the preconditioner-dependent optimal space for the balancing correction, a 2×2 example in which the Rayleigh objective selects the wrong direction, and a spectrum-level illustration on a held-out operator;
2. controlled interventions (an objective swap, targets crossed between two preconditioners, and two architectures) that isolate solver alignment from architecture and optimization quality; and
3. experiments that separate target quality from prediction and from cost, against *target reuse* (deploying the nearest stored training target, or an interpolation of four, instead of a prediction), an extra cycle, strong classical baselines, distribution shift, flops, and measured time, reporting every family we tried, including those that failed.

2 Related work

Multigrid and adaptive coarse spaces. Classical AMG builds interpolation from matrix-dependent strength (Ruge & Stüben, 1987); non-grid-aligned anisotropy is a known difficult case (Trottenberg et al., 2000), and evolution-based strength measures target algebraic smoothness more robustly (Olson et al., 2010). We also test HYPRE's production BoomerAMG (Henson & Yang, 2002). GenEO selects coarse modes from local generalized eigenproblems (Spillane et al., 2013); our GenEO controls insert such spaces around the same AMG preconditioner. Relaxing random test vectors to expose algebraically smooth error is established in adaptive and bootstrap AMG (Brezina et al., 2006; Brandt et al., 2011), and AMG theory, compatible relaxation, and optimal interpolation tie coarse-space quality to the error left by relaxation rather than to the operator spectrum alone (Falgout & Vassilevski, 2004; Brannick et al., 2017; Xu & Zikatanov, 2017). Deflation and augmentation theory relates subspace choice to the preconditioned spectrum (Nicolaides, 1987; Tang et al., 2009; Gaul et al., 2013; Kahl & Rittich, 2012). We claim none of these principles as new; Lemma 1 restates the dependence for our correction.

Learned multilevel solvers. Learned methods predict smoothers, interpolation, coarse spaces, or sparse factors (Greenfeld et al., 2019; Luz et al., 2020; Häusner et al., 2024; Booth et al., 2024; Kopaničáková & Karniadakis, 2025; Kopaničáková et al., 2025; Levrero-Florencio et al., 2026; Benanti et al., 2026; Liu et al., 2026). RAPNet predicts sparse AMG coarse operators with a GNN (Fink et al., 2026), and Klawonn et al. (2026) predict adaptive domain-decomposition spaces from eigensolve-derived supervision. The closest architecture is NeurKItt (Luo et al., 2024), which predicts an orthonormal subspace with an FNO trained against exact invariant subspaces; our Rayleigh FNO instead uses an unsupervised Rayleigh objective. NeuraLSP learns fixed-rank coarse bases from relaxed random test vectors (Benanti et al., 2026), and Melchers et al. (2026) ask when a neural operator can replace a coarse solve, whereas we predict a basis for an unchanged correction. Our solver-aligned objective uses approximate targets generated by the fixed preconditioner, so it is distillation, not label-free training. Neither neural basis prediction nor supervision by relaxed error is our claimed novelty; we hold deployment fixed and separate target quality, prediction, and cost through matched interventions. Appendix C.3 gives a matched NeurKItt-style supervision control.

Rayleigh quotients are also standard for learning spectral subspaces (Pfau et al., 2018; Gemp et al., 2020; Deng et al., 2022); we test whether the exact minimizer of such an objective matches the directions a multigrid preconditioner reduces slowly.

3 METHOD AND EVALUATION

3.1 ONE CORRECTION, TWO TRAINING OBJECTIVES

The two objectives differ only in how they choose the coarse space; the deployed correction is identical. We discretize

$$-\nabla\cdot(K(\mathbf{s})\nabla u(\mathbf{s})) = f(\mathbf{s}) \quad \Longrightarrow \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \tag{1}$$

where $\mathbf{A}$ is symmetric positive definite, with cell-centered finite volumes on the unit square or cube and homogeneous Dirichlet boundaries. The metric is the number of preconditioned CG iterations until the recursive relative residual reaches 10^{-8}.

The correction. Let M_1 be the base preconditioner, one multigrid V-cycle, and let $\mathbf{Z} \in \mathbb{R}^{N\times m}$ span the coarse space. With $E = \mathbf{Z}^\top\mathbf{A}\mathbf{Z}$, the classical balancing correction (Nicolaides, 1987; Mandel, 1993; Tang et al., 2009) is

$$M_{\mathrm{c}} = \mathbf{Z}E^{-1}\mathbf{Z}^\top + (I - \mathbf{Z}E^{-1}\mathbf{Z}^\top\mathbf{A})M_1(I - \mathbf{A}\mathbf{Z}E^{-1}\mathbf{Z}^\top). \tag{2}$$

The multigrid hierarchy defining M_1 is kept, the $m \times m$ matrix E is factorized once per operator, and every corrected method uses this implementation. "One cycle" denotes M_1 and "two cycles" two applications of it.

The Rayleigh objective. A Fourier neural operator (FNO) Φ_θ maps operator channels and coordinates to m vectors $V \in \mathbb{R}^{N\times m}$; a thin QR gives $V = \mathbf{Q}R$. The Rayleigh training objective is

$$\mathcal{L}_{\mathrm{RQ}}(\theta;\mathbf{A}) = \frac{1}{m}\,\mathrm{tr}\Big(\mathbf{Q}^\top\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}\mathbf{Q}\Big), \qquad \mathbf{D} = \mathrm{diag}(\mathbf{A}), \tag{3}$$

which needs no labels. Its exact minimizer, the *exact Rayleigh space*, is the bottom eigenspace of $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. At deployment either objective sets $\mathbf{Z} = \mathrm{qr}(\mathbf{D}^{-1/2}V)$; this QR changes the basis of the mapped span, not the span.

Why should the target depend on M_1? A 2×2 example shows the mismatch. Let $\mathbf{A}q_\pm = (1 \pm \rho)q_\pm$ with $\mathrm{diag}(\mathbf{A}) = I$ and $0 < \rho, \epsilon < 1$, and let $M_1\mathbf{A}q_- = q_-$ and $M_1\mathbf{A}q_+ = \epsilon q_+$. Rayleigh minimization selects q_-, a direction M_1 already solves exactly; correcting it leaves condition number $1/\epsilon$. Correcting the higher-Rayleigh vector q_+ gives condition number one. In general, let $H_1 = \mathbf{A}^{1/2}M_1\mathbf{A}^{1/2}$ and let P be the orthogonal projector onto $\mathbf{A}^{1/2}\,\mathrm{range}(\mathbf{Z})$. Then

$$\mathbf{A}^{1/2}M_{\mathrm{c}}\mathbf{A}^{1/2} = P + (I - P)H_1(I - P). \tag{4}$$

Lemma 1 (known; restated for balancing)**.** *Let M_1 be SPD with H_1 having spectrum in $(0, 1]$, and $1 \le m < N$. A space $\mathrm{range}(\mathbf{A}^{1/2}\mathbf{Z})$ minimizing the condition number of $\mathbf{A}^{1/2}M_{\mathrm{c}}\mathbf{A}^{1/2}$ can be chosen as a bottom-m invariant subspace of H_1.*

This is the balancing analogue of optimal-interpolation and deflation results (Xu & Zikatanov, 2017; Brannick et al., 2017; Tang et al., 2009); the proof is in Appendix B.1. In words: spend the limited rank on the directions the deployed preconditioner reduces most slowly, not on the lowest-energy directions of the operator. Equation (4) also shows that, when $H_1 \preceq I$, every eigenvalue of the corrected operator lies in $[\lambda_{\min}(H_1), 1]$, so no choice of $\mathbf{Z}$ can make the condition number exceed $1/\lambda_{\min}(H_1)$, the uncorrected value when $\lambda_{\max}(H_1) = 1$. Iteration counts are not bounded this way and can still rise (Section 4.4). The premise $H_1 \preceq I$ holds in exact arithmetic for both multigrid preconditioners we correct: the PyAMG V-cycle with symmetric Gauss–Seidel and the geometric V-cycle with weighted Jacobi, each with an exact coarsest solve (Appendix B.1). The argument does not cover BoomerAMG, whose relaxation differs, or finite precision.

The solver-aligned objective. Lemma 1 motivates a target tied to M_1. Starting from a fixed Gaussian rank-m matrix Ω, we apply the error propagator of the base preconditioner,

$$\widehat{Z} = \mathrm{orth}\big[(I - M_1\mathbf{A})^{64}\Omega\big], \qquad \mathcal{L}_{\mathrm{SA}} = 1 - \frac{1}{m}\|Z_\theta^\top\widehat{Z}\|_F^2, \quad Z_\theta = \mathrm{qr}(\mathbf{D}^{-1/2}V), \tag{5}$$

reorthonormalizing after every application. We call $\widehat{Z}$ the *finite solver target*; each label costs 64 rank-32 block iterations, 2,048 single-vector V-cycle applications, so training is distillation rather than label-free (glossary in Table 4).

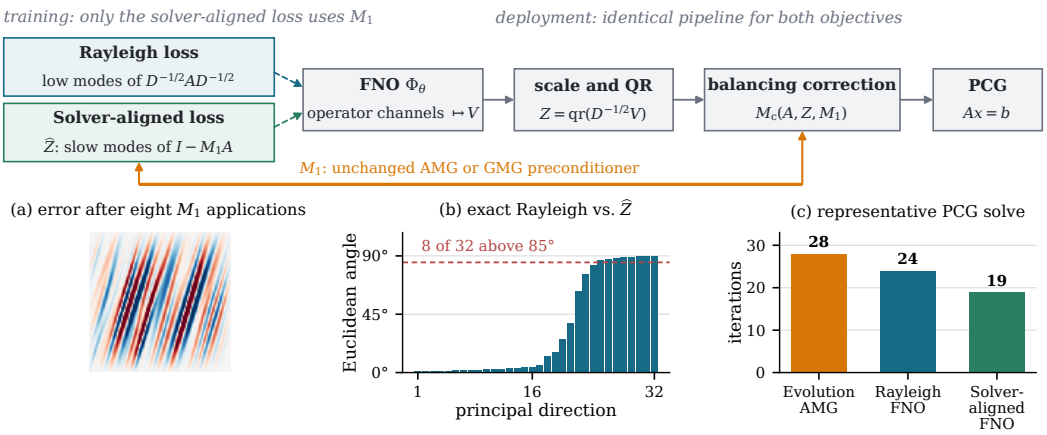

Figure 1: Training, deployment, and solver response. Only the solver-aligned objective uses the base preconditioner M_1 during training; deployment is identical. On the held-out operator closest to the Objective-suite means, M_1 leaves structured error (a), the exact Rayleigh space and finite solver target differ by eight Euclidean principal angles above $85°$ (b), and the predicted-space correction reduces the one-right-hand-side PCG count (c). Sections 4.1 and 4.2 report aggregate results.

Why the finite target approximates the ideal space. One application of M_1 maps an error e to $(I - M_1\mathbf{A})e$. Because $\mathbf{A}^{1/2}(I - M_1\mathbf{A})\mathbf{A}^{-1/2} = I - H_1$, the error component along an eigenvector of $M_1\mathbf{A}$ with eigenvalue μ shrinks by the factor $1 - \mu$ per cycle: directions that M_1 removes well (μ near one) vanish within a few cycles, while those it removes slowly (μ near zero) survive and set the condition number. Equation (5) is therefore subspace iteration on the error propagator: repeated cycles applied to random vectors keep only the slowest directions, whose span, the optimal space of Lemma 1, we call the *ideal solver space* (Figure 1a shows the surviving error). In the 2×2 example the first step annihilates q_- and keeps q_+, exactly the direction the Rayleigh objective discards. The Rayleigh objective sees only $\mathbf{A}$, so it agrees with the ideal space only when the modes M_1 handles worst are also the lowest-energy ones, nearly the case around default AMG (Section 4.2) but not around Evolution AMG. Directions near the rank-m cutoff converge slowly, so 64 steps resolve them only partly; hence a *finite* target (on the online-construction suite, 16, 32, and 64 steps give 23.29, 21.23, and 20.31 mean iterations; Section 4.4 and Appendix B.2).

3.2 Architecture and Training

The 2D FNO has four spectral layers, width 64, 16 modes per axis, rank 32, and 2.41 million parameters; it receives the three tensor entries and two coordinates. The Rayleigh and solver-aligned FNOs train for 80 epochs on the same 200 operators, and only the loss and its offline labels differ. The 3D FNO has the same depth and modes, rank 16, and scalar conductivity plus three coordinates as input; we use a Large (37.8M-parameter) and a Compact (1.33M-parameter) version, trained for 250 epochs on 64 operators (a 200-operator Large Rayleigh FNO is a data-size control). All models use Adam at 10^{-3}, gradient clipping at 1, and cosine decay. Generating the 200 2D training labels costs 409,600 single-vector V-cycle applications offline. Appendix A gives the input channels, checkpoint history, and exact label accounting.

3.3 Benchmarks, Suites, and Reporting

Default AMG and Evolution AMG use the same PyAMG Ruge–Stüben hierarchy with classical or evolution strength (Bell et al., 2022); BoomerAMG is HYPRE's implementation; threshold-0.5 BoomerAMG uses strong threshold 0.5, chosen on separate operators. GMG3D is our geometric V-cycle for the 3D family. Appendix A gives the exact configurations.

Rotated anisotropy (2D). Each operator has $K = R(\theta)\,\mathrm{diag}(k_1, k_2)R(\theta)^\top$ with $\theta \in [16, 25]°$ and $k_1/k_2 \in [200, 5000]$, on a 256×256 grid. The narrow angular band, the *training band*, was chosen during exploration because default AMG fails sharply there; it is a controlled stress test, not a representative prior. The base preconditioner is Evolution AMG.

Table 1: Stronger classical preconditioners sharply reduce the apparent learned gain. Baseline suite: rotated anisotropy at $n = 256$, 12 held-out operators. Each row is a matched comparison around one preconditioner. Default AMG, GMG, and Evolution AMG use one right-hand side per operator; BoomerAMG averages 16. GMG is a geometric V-cycle. The first three rows use the first Rayleigh checkpoint, which also received an input channel absent from the loss; the BoomerAMG row uses a retrained checkpoint without it (Appendix A.1). Lower is better.

Preconditioner	Uncorrected	With Rayleigh FNO	Aggregate reduction
Shared checkpoint and one right-hand side per operator			
Default AMG	106.5	55.4	48.0%
GMG	46.7	43.1	7.7%
Evolution AMG	32.6	26.6	18.4%
Retrained checkpoint and 16 right-hand sides			
BoomerAMG	10.04	9.63	4.1%

3D obstacles. Near-impermeable barrier planes with random openings force nonlocal paths through a scalar coefficient field. We train at $n = 48$ with rank 16 around GMG3D and test zero-shot transfer to larger grids, Bentheimer sandstone scans, eight public trabecular-bone volumes, and scalarized SPE10 coefficients (a fixed-contrast diffusion proxy, not a reservoir simulation).

Suites and reporting. Each comparison uses its own independently generated suite of held-out operators, named by its question (Objective, Crossed, Retrieval, and so on; Appendix A.1); rows from different suites are never pooled, and within a suite all methods share operators, right-hand sides, and tolerance. We report equal-operator mean iterations and the mean per-operator iteration reduction

$$\frac{100}{J}\sum_{j=1}^{J}\left(1-\frac{k_j^{\text{method}}}{k_j^{\text{base}}}\right), \tag{6}$$

where k_j is the mean count over right-hand sides for operator j; higher is better. Intervals are descriptive over the fixed held-out operators.

4 Results

Section 4.1 fixes the base preconditioner. Sections 4.2 and 4.3 test the target requirement, Sections 4.4–4.6 the prediction requirement, and Section 4.7 the cost requirement.

4.1 Setup: strong classical baselines set the bar

How much of a learned gain survives a stronger base preconditioner? Less than half. Against default AMG, the Rayleigh FNO correction reduces the aggregate mean iteration count by 48.0%; against Evolution AMG by 18.4%, and against BoomerAMG by only 4.1% (Table 1). Strength is not the whole story: GMG, a 2D geometric V-cycle, is weaker than Evolution AMG yet gains only 7.7%, because a space learned for one preconditioner's error need not suit another's (Section 4.2). We therefore use Evolution AMG, the strongest preconditioner the correction still improves clearly, as M_1 for all 2D target experiments, and compare against BoomerAMG in total time in Section 4.7. The 18.4% here is a reduction in aggregate means on the 12-operator Baseline suite; the 17.6% in Section 4.2 is a mean per-operator reduction on a different suite. A finite solver target built around BoomerAMG lowers its mean from 11.05 to 9.50 iterations on separate operators, but two cycles reach 7.47, so we trained no network for it (Appendix B.2). Training-seed variation, paired intervals, and a separately tuned BoomerAMG control are in Appendix C.

4.2 Target: the useful space depends on the base preconditioner

Does training against the error M_1 leaves behind give a better coarse space than the Rayleigh objective? Yes. We hold the FNO architecture, rank, training operators, optimizer schedule, deployment map, and Evolution AMG fixed, and change only the training signal. On the Objective suite, the

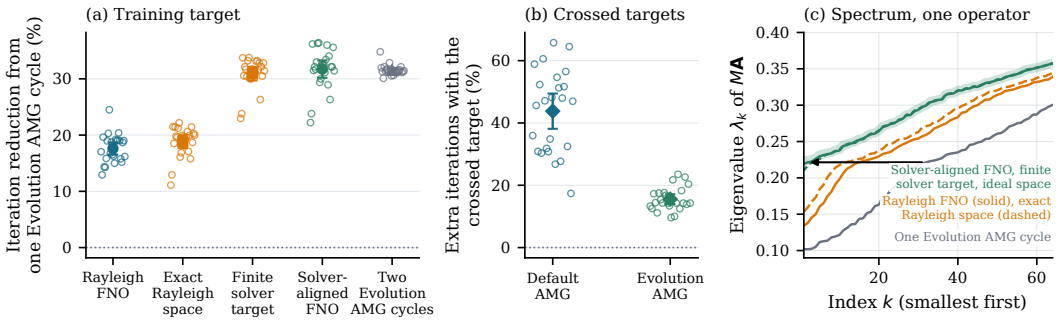

Figure 2: The useful target depends on the base preconditioner. (a) Objective suite, 24 operators: per-operator iteration reduction from one Evolution AMG cycle (open circles, means over eight right-hand sides) with equal-operator means and descriptive 95% intervals; higher is better. Exact spaces and targets are references, not learned deployments. (b) Crossed suite: extra iterations, in percent of the matched-target count, when the target generated by the other preconditioner is deployed; positive values favor the matched target. (c) The 64 smallest eigenvalues of $M\mathbf{A}$ on the Figure 1 operator, where M is Evolution AMG alone or with the rank-32 correction built from each space (double precision). Each correction moves 32 eigenvalues to one (off the plot). The ideal solver space removes exactly the 32 smallest eigenvalues of Evolution AMG (arrow); the finite solver target and solver-aligned FNO nearly coincide with it.

solver-aligned FNO reduces iterations by 31.73% per operator, against 17.61% for the Rayleigh FNO and 18.82% for the exact Rayleigh space (Figure 2a): the solver-aligned FNO beats even the exact Rayleigh space (the finite solver target itself averages 21.62 iterations against 25.43 for the exact Rayleigh space), and the solver-aligned FNO is better than both Rayleigh variants on every operator. In iterations the gain equals one extra cycle: the solver-aligned FNO averages 21.26 against 21.63 for two Evolution AMG cycles on this suite, and it also ties two cycles on the Architecture suite (19.33 versus 19.27) and in the online-construction comparison of Appendix B.2 (20.22 versus 20.21).

The spectrum shows why (Figure 2c); lifting the smallest eigenvalues of the preconditioned operator lowers its condition number, and CG iterations grow roughly with the square root of the condition number. On the held-out operator of Figure 1, the ideal solver space removes the 32 smallest eigenvalues of $M_1\mathbf{A}$, and the finite solver target and the solver-aligned FNO nearly coincide with it (smallest remaining eigenvalue 0.210 and 0.220 against 0.223; the network slightly exceeds the finite solver target, which is itself only an approximation of the ideal space). The exact Rayleigh space, although optimal for its own objective, lifts the bottom of the spectrum much less (0.154), and the Rayleigh FNO less still (0.134), from 0.102 for one cycle. Around default AMG the picture reverses: there the exact Rayleigh space is almost the ideal solver space (Appendix B.4), which explains why Rayleigh training looks strong against default AMG in Table 1 and weak against a stronger preconditioner.

If the useful target depends on M_1, a target built for one preconditioner should lose when deployed around another. It does. Targets generated by default AMG and by Evolution AMG from the same initial block and construction length are crossed between the two preconditioners on the Crossed suite: the matched target wins on all 24 operators for both (Figure 2b), stably across three hierarchy realizations (Appendix B.6). Around the weaker default AMG the matched target is worth more than an extra cycle ($105.70 \rightarrow 69.11$ against 72.92 for two cycles); around Evolution AMG it is worth about one.

4.3 Target: the effect is not specific to the FNO

Is the gain an FNO artifact? No. A parameter-matched U-Net and the FNO are each trained with both objectives on the same 200 operators, rank, optimizer schedule, diagonal scaling, QR map, and Evolution AMG. On the Architecture suite, changing the objective lowers the mean by 3.91 iterations for the FNO and 3.47 for the U-Net, improving every operator in both comparisons, while changing the architecture has a much smaller effect (Table 2). This supports solver alignment across

Table 2: Changing the objective helps both architectures more than changing between them. Architecture suite: 24 operators, eight right-hand sides each. Entries are mean iterations with the mean per-operator reduction from one Evolution AMG cycle (28.11 iterations) in parentheses. The last column is Rayleigh minus solver-aligned iterations with a descriptive 95% interval.

Architecture	Rayleigh	Solver-aligned	Objective difference
FNO	23.23 (16.58%)	19.33 (30.06%)	3.91 [3.18, 4.63]
U-Net	23.42 (15.99%)	19.95 (27.75%)	3.47 [2.71, 4.23]

these two models, not a claim that architecture is irrelevant in general (Appendix A.3). Within this 2D family the evidence therefore favors the solver-aligned target; the harder question is whether a network can deliver that target on operators it has not seen.

4.4 Prediction: it works only inside the training band

Can the network predict the target on new operators? Inside the training band it does, slightly better than reusing a stored target; outside the band and on a second 2D family it does not. On the Retrieval suite, the solver-aligned FNO (mean of three independently initialized networks) averages 20.18 iterations, against 20.62 for the nearest training target and 20.46 for a four-neighbor interpolation of training targets; both paired intervals exclude zero, while the interval against two Evolution AMG cycles (19.97) includes zero (Appendix B.2). On a further suite, the FNO also matches a target built online by 64 block iterations (20.22 against 20.31 iterations) without that construction's 2,048 V-cycle applications per operator; 16 block iterations reach only 23.29. On the Disjoint-angle suite, with angles in $[27, 36]^\circ$ fully outside the training band $[16, 25]^\circ$, every method needs fewer iterations than in band, and the FNO (18.97) loses to target reuse (18.13 nearest, 18.27 interpolated) and to two cycles (13.63); it remains slightly better than one cycle (20.08) (Appendix B.7).

A second 2D family shows the same bottleneck earlier. With lognormal coefficients of contrast 10^4, the finite solver target improves Evolution AMG from 21.91 to 13.45 iterations on eight separate operators, so a useful target exists. The solver-aligned FNO trained on this family, however, is worse than the uncorrected preconditioner on all 12 validation operators (22.95 against 21.34, a -7.91% reduction). Networks trained with energy-norm and worst-direction losses also stay worse than uncorrected (20.25 and 20.02 against 19.80, re-evaluated with one shared hierarchy per operator, which changes the uncorrected mean). A worse iteration count does not contradict the bound of Section 3.1: that bound concerns the condition number in exact arithmetic, while the deployed correction uses single-precision coarse arrays and CG iterations also depend on how the spectrum is distributed (Appendix D.3). Appendix B.8 lists this and every other family we tried, including two that stopped before any training.

4.5 Prediction: in 3D a useful target exists but is not predicted

Does the 3D obstacle family have a useful target, and is it predicted? The target exists, but no tested predictor recovers it. We correct GMG3D because one Ruge–Stüben AMG cycle already needs only 7.82 iterations on a separate nine-operator suite from this family (Appendix D), far fewer than any GMG3D variant below; the 3D study is therefore a test of the mechanism, not a solver claim. Around one GMG3D cycle, the finite solver target lowers the mean from 24.75 to 14.33 iterations on the six-operator 3D target suite, better than two cycles (17.40; Figure 3). Three Compact FNOs trained on the same 64 operators with the Rayleigh loss, the solver-aligned loss, and an energy-norm version of it average 22.48, 21.56, and 22.73. The Large FNOs do better but not well enough: the Rayleigh FNO is the best predictor at 18.04, the solver-aligned FNO reaches 18.81, and Large FNOs trained with the mean and the worst-direction energy-norm losses give 18.15 and 18.10. No Compact candidate met the criterion fixed in advance for a confirmation run, so no further 3D test set was opened (Appendix B.5). This is a failure of the tested predictors, not evidence that the target cannot be learned by another model. On Bentheimer sandstone scans the target comparison itself is mixed: at ranks 16, 32, and 64 the exact Rayleigh space cuts GMG3D iterations by 24.5%, 71.7%, and 77.6%, and finite solver targets after 40 block iterations (not the 64 of Eq. (5)) by 7.2%, 72.6%, and 80.3% (Appendix D.3).

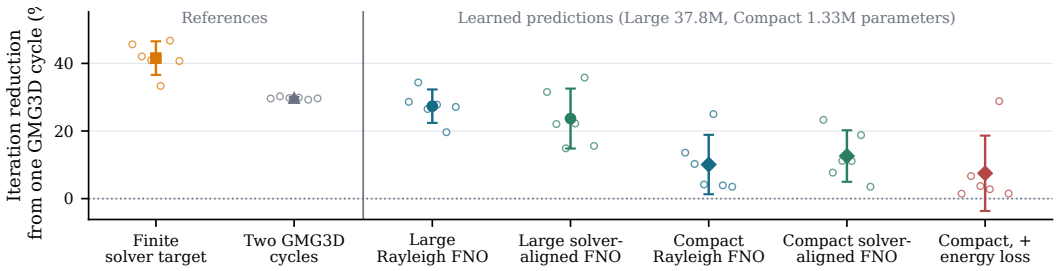

Figure 3: In 3D the target is useful but not predicted. Six development obstacle operators, eight right-hand sides each: per-operator iteration reduction from one GMG3D cycle (open circles) with equal-operator means and descriptive 95% intervals; higher is better. References (left) are not learned deployments. Large and Compact FNOs have 37.8M and 1.33M parameters.

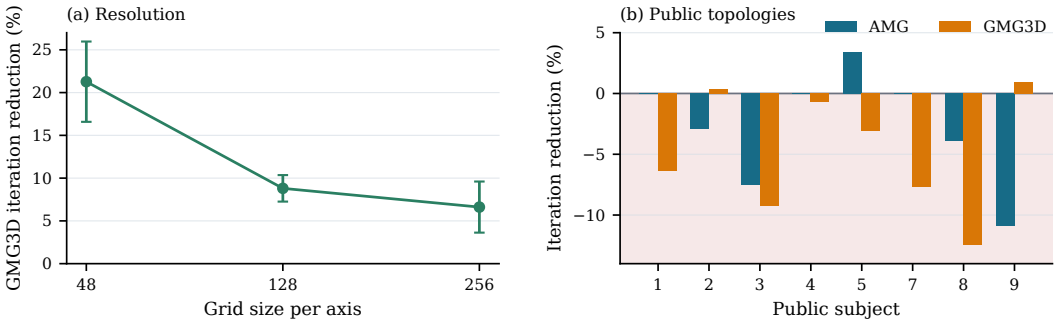

Figure 4: The rank-16 Rayleigh FNO weakens with resolution and fails on public topologies. (a) Resolution transfer over three held-out geometries per size; bars are sample standard deviations. (b) Per-subject transfer on eight public topologies; bars below zero indicate that adding the learned correction increases iterations. Subject identifiers follow the public dataset record; geometries differ from Figure 3.

4.6 Prediction: resolution and topology shift break transfer

Does a trained 3D checkpoint transfer to larger grids and real geometry? Weakly across resolution, and not across topology. We test the rank-16 Rayleigh FNO trained at $n = 48$ on 64 operators, the first 3D checkpoint we trained. Averaged over three training runs and three geometries per size, its mean GMG3D iteration reduction falls from 21.3% at $n = 48$ to 8.8% at $n = 128$ and 6.6% at $n = 256$ (Figure 4a), and one AMG cycle still needs fewer iterations at every size. On eight public trabecular-bone volumes the correction does not improve either preconditioner on average and worsens GMG3D on six subjects (Figure 4b). The SPE10 proxy is essentially unchanged, a grouped sandstone study is unfavorable, and HYPRE Example 7 is inconsistent across cases (Appendices D.5–D.7).

4.7 Cost: iteration gains versus total time

Does the correction beat strong classical alternatives in total time? No. It is faster than two Evolution AMG cycles only in our Python implementation, costs more flops per iteration than two cycles, and is $2.6\times$ slower than BoomerAMG (Table 13 and Figure 5 in Appendix B.3). All times include hierarchy construction, inference, QR, coarse factorization, and every solve. On the Timing suite at 64 right-hand sides, the solver-aligned FNO takes 14.26 seconds per operator, 37.0% less than two Evolution AMG cycles (22.62) and 15.2% less than one cycle (16.81), but it is not separated from target reuse; absolute times differ between suites, which use different operators and hierarchy seeds. On the Common-hardware suite, default and threshold-0.5 BoomerAMG take 6.82 and 7.39 seconds, against 17.85 for the FNO and 18.35 for Evolution AMG, where the FNO's margin over one cycle shrinks to 2.7%.

A flop count, which does not depend on the implementation, explains both results. The correction adds $8mN = 256N$ flops per application, more than a V-cycle: on the Common-hardware operators, one Evolution AMG cycle costs about $196N$ (operator complexity 2.05) and one BoomerAMG cycle about $208N$ (Appendix B.3). A corrected iteration in this balancing form therefore costs more than two cycles ($470N$ against $411N$ flops including the matrix-vector product; the A-DEF2 form with the same spectrum (Tang et al., 2009) would cost about as much as two cycles), and per solve the corrected preconditioner costs about $9.2 \times 10^3 N$ flops against $6.1 \times 10^3 N$ for one cycle and $2.3 \times 10^3 N$ for BoomerAMG, which needs about ten iterations. The time advantage over two cycles is thus an implementation effect of a few dense single-precision products against level-by-level sparse multigrid in Python, not an algorithmic one, while BoomerAMG wins algorithmically. BoomerAMG is also compiled HYPRE, whereas our pipeline runs in Python and SciPy; this asymmetry favors BoomerAMG in time but does not create its advantage in iterations or flops. Building the target online by plain block iteration needs 64 steps, 2,048 V-cycle applications, to match the FNO (32 steps reach only 21.23 iterations; Appendix B.2), more than the roughly 1,300 cycle applications of all 64 solves; accelerated eigensolvers could be cheaper. Krylov recycling is the obvious classical way to amortize work over many right-hand sides; we did not test it.

5 Discussion

Target. The evidence for solver alignment is interventional within its scope: with everything else fixed, the finite solver target beats the Rayleigh target, beats even the Rayleigh objective's exact optimum, and loses when crossed to another preconditioner. In the 2D family, for Evolution AMG and both networks we tested, the operator-only objective is the wrong target for a correction that sits on top of a strong preconditioner.

Prediction. Prediction, not the target, is the bottleneck. Inside the training band the network is only slightly better than reusing a stored target, and outside it, on a second 2D family, and in 3D it does not recover the target's benefit. In 3D even the objective swap did not reproduce: the Large Rayleigh FNO (18.04) edged out its solver-aligned counterpart (18.81). Energy-norm and worst-direction losses did not close the gap in the lognormal or 3D families. Appendix diagnostics show that a few missed directions explain part, but not all, of the 3D gap.

Cost. Iteration reduction diagnoses the mechanism but does not establish acceleration (Section 4.7). The learned correction ties a second cycle of the same hierarchy in iterations and is faster in time only because dense products are cheap in our implementation, while a strong production AMG leaves little slow error for any correction to remove. A learned space is most plausible where building the target online is expensive and one prediction serves many right-hand sides.

A checklist for learned coarse spaces. Our results suggest four tests for any learned correction: condition the training target on the deployed preconditioner; compare with the exact optimum of one's own loss, not only with a trained model; cross targets between preconditioners; and compare against target reuse, an extra cycle, flops, and total time around a strong baseline.

6 Limitations and conclusion

Positive prediction evidence is limited to interpolation within one controlled 2D family, whose angle band and base preconditioner (Evolution AMG) were chosen after seeing default AMG fail there. The 3D operators are synthetic or scalar-diffusion versions of public images, and no 200-operator 3D solver-aligned model was trained. Timing is single-host.

A learned coarse space must meet three requirements. The target requirement holds in the 2D family, where training against the error the base preconditioner leaves behind beats the Rayleigh objective and moves with the preconditioner; a useful target also exists in 3D. Prediction, which holds only inside the training band, is the bottleneck; even there the correction is worth about one extra cycle in iterations while costing more flops, so BoomerAMG remains faster. Because the finite solver target can be computed offline, a predictor should be judged by how much of the target's benefit it recovers, against target reuse and an extra cycle around a strong baseline: in 2D the solver-aligned FNO recovers essentially all of it (21.26 against 21.62 iterations), in 3D the best predictor recovers under two-thirds ($24.75 \to 18.04$ against 14.33).

AI USE STATEMENT

We used generative AI tools to develop and refine research ideas and hypotheses, formulate and check mathematical claims and proofs, design experiments, generate synthetic 2D and 3D operators, prepare public datasets, implement methods, and interpret results. Translation and qualitative or thematic analysis were not applicable. We also used AI for brainstorming and literature search, writing and editing code and artifacts, creating scientific figures, drafting and editing the paper, and reviewing artifacts. We reviewed all AI-assisted work. Reported numbers are automatically checked against released records; released validators and unit tests pass; protocol records and a chronology audit distinguish analyses fixed before evaluation from post hoc ones; and citations were checked against Semantic Scholar records. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies numerical linear solvers and conducts no human-subject research. We access only public numerical fields and segmentation masks, not identifying metadata, and use them only to construct numerical operators. We are not aware of additional ethical concerns.

REPRODUCIBILITY STATEMENT

The anonymous repository `https://anonymous.4open.science/r/learned-coarse-spaces-9AEC` contains source code, per-instance records, protocols, seeds, and commands for reproducing the reported experiments, including the negative results in Appendix B.8. It includes the 2D and compact 3D checkpoints; the larger 3D checkpoints are listed by hash. Independent numerical tests cover operator assembly, PCG, multigrid cycles, coarse correction, and the GenEO eigenproblem. Appendix E documents the public data and licenses.

REFERENCES

Nathan Bell, Luke N. Olson, and Jacob Schroder. Pyamg: Algebraic multigrid solvers in python. *Journal of Open Source Software*, 7(72):4142, 2022. doi: 10.21105/joss.04142. URL https://doi.org/10.21105/joss.04142.

Alexander Benanti, Xi Han, and Hong Qin. Neuralsp: A neural spectral preconditioner for accelerating pde solvers, 2026. URL https://arxiv.org/abs/2601.20174.

Joshua Dennis Booth, Hongyang Sun, and Trevor Garnett. Neural acceleration of incomplete Cholesky preconditioners. *arXiv preprint arXiv:2403.00743*, 2024. URL https://api.semanticscholar.org/CorpusID:268230357.

Achi Brandt, James J. Brannick, Karsten Kahl, and Irene Livshits. Bootstrap amg. *SIAM J. Sci. Comput.*, 33:612–632, 2011. URL https://api.semanticscholar.org/CorpusID:14724638.

James J. Brannick, Fei Cao, Karsten Kahl, Robert D. Falgout, and Xiaozhe Hu. Optimal interpolation and compatible relaxation in classical algebraic multigrid. *SIAM J. Sci. Comput.*, 40, 2017. URL https://api.semanticscholar.org/CorpusID:51880165.

M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive algebraic multigrid. *SIAM Journal on Scientific Computing*, 27(4):1261–1286, 2006. doi: 10.1137/040614402. URL https://doi.org/10.1137/040614402.

Chen Cai, Nikolaos N. Vlassis, Lucas Magee, Ran Ma, Zeyu Xiong, Bahador Bahmani, Teng-Fong Wong, Yusu Wang, and Waiching Sun. Equivariant geometric learning for digital rock physics: estimating formation factor and effective permeability tensors from morse graph. *ArXiv*, abs/2104.05608, 2021. URL https://api.semanticscholar.org/CorpusID:233209670.

M. A. Christie and M. J. Blunt. Tenth spe comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Evaluation & Engineering*, 4(04):308–317, 08 2001. ISSN 1094-6470. doi: 10.2118/72469-PA. URL https://doi.org/10.2118/72469-PA.

Zhijie Deng, Jiaxin Shi, and Jun Zhu. NeuralEF: Deconstructing kernels by deep neural networks. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:248496652.

Robert D. Falgout and Panayot S. Vassilevski. On generalizing the algebraic multigrid framework. *SIAM J. Numer. Anal.*, 42:1669–1693, 2004. URL https://api.semanticscholar.org/CorpusID:4482770.

Yali Fink, Ido Ben-Yair, Lars Ruthotto, and Eran Treister. Rapnet: Accelerating algebraic multigrid with learned sparse corrections. *ArXiv*, abs/2605.26854, 2026. URL https://api.semanticscholar.org/CorpusID:288671950.

André Gaul, Martin H. Gutknecht, Jörg Liesen, and Reinhard Nabben. A framework for deflated and augmented krylov subspace methods. *SIAM Journal on Matrix Analysis and Applications*, 34(2):495–518, 2013. doi: 10.1137/110820713. URL https://doi.org/10.1137/110820713.

Ian M. Gemp, Brian McWilliams, Claire Vernade, and Thore Graepel. Eigengame: Pca as a nash equilibrium. *ArXiv*, abs/2010.00554, 2020. URL https://api.semanticscholar.org/CorpusID:222090711.

Daniel Greenfeld, Meirav Galun, Ronen Basri, Irad Yavneh, and Ron Kimmel. Learning to optimize multigrid pde solvers. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:67855720.

Paul Häusner, Ozan Öktem, and Jens Sjölund. Neural incomplete factorization: learning preconditioners for the conjugate gradient method. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FozLrZ3CI5.

Van Emden Henson and Ulrike Meier Yang. Boomeramg: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41:155–177, 2002. URL https://api.semanticscholar.org/CorpusID:13981261.

Samuel J. Jackson, Yufu Niu, Sojwal Manoorkar, Peyman Mostaghimi, and Ryan T. Armstrong. Multi-resolution x-ray micro-ct images of bentheimer sandstones, 2021.

Samuel J. Jackson, Yufu Niu, Sojwal Manoorkar, Peyman Mostaghimi, and Ryan Troy Armstrong. Deep learning of multiresolution x-ray micro-computed-tomography images for multiscale modeling. *Physical Review Applied*, 2022. URL https://api.semanticscholar.org/CorpusID:249141305.

Karsten Kahl and Hannah Rittich. The deflated conjugate gradient method: Convergence, perturbation and accuracy. *arXiv: Numerical Analysis*, 2012. URL https://api.semanticscholar.org/CorpusID:119175817.

Axel Klawonn, Martin Lanser, and Janine Weber-Hamacher. Learning adaptive coarse spaces using transferable neural network models for linear and nonlinear overlapping domain decomposition methods. *ArXiv*, abs/2607.06261, 2026. URL https://api.semanticscholar.org/CorpusID:289942000.

Alena Kopaničáková and George Em Karniadakis. DeepONet based preconditioning strategies for solving parametric linear systems of equations. *SIAM Journal on Scientific Computing*, 47(1): C151–C181, 2025. doi: 10.1137/24M162861X. URL https://api.semanticscholar.org/CorpusID:266756024.

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, 2025. URL https://api.semanticscholar.org/CorpusID:280417448.

Michael Kuczynski. Microct trabecular bone samples for trabecular thickness and separation measures, 2024.

Michael T. Kuczynski, Nathan J. Neeteson, Kathryn S. Stok, Andrew J. Burghardt, Michelle A. Espinosa Hernandez, Jared Vicory, Justin J. Tse, Pholpat Durongbhan, Serena Bonaretti, Andy Kin On Wong, Steven K. Boyd, and Sarah L. Manske. Ormir_xct: A python package for high resolution peripheral quantitative computed tomography image processing. *J. Open Source Softw.*, 9:6084, 2023. URL https://api.semanticscholar.org/CorpusID:261682219.

Francesc Levrero-Florencio, Youngkyu Lee, Jay Pathak, and George Em Karniadakis. NS-POD: Accelerating krylov solvers via DeepONet-learned POD subspaces. *arXiv preprint arXiv:2605.07828*, 2026. URL https://api.semanticscholar.org/CorpusID:288147561.

Yucheng Liu, Tak-Shing Au Yeung, Eric T. Chung, and Simon See. Graph-based meshfree multiscale coarse space approximation for two-level schwarz methods. *ArXiv*, abs/2603.26149, 2026. URL https://api.semanticscholar.org/CorpusID:286942016.

Éverton Lucas-Oliveira, Mariane Barsi-Andreeta, Rodrigo F. Neumann, Willian A. Trevizan, Mathias B. Steiner, and Tito J. Bonagamba. Micro-computed tomography of sandstone rocks: Raw, filtered and segmented datasets. *Data in Brief*, 41, 2022. URL https://api.semanticscholar.org/CorpusID:246509047.

Jian Luo, Jie Wang, Hong Wang, Huanshuo Dong, Zijie Geng, Hanzhu Chen, and Yufei Kuang. Neural krylov iteration for accelerating linear system solving. *Advances in Neural Information Processing Systems 37*, 2024. URL https://api.semanticscholar.org/CorpusID:276318500.

Ilay Luz, Meirav Galun, Haggai Maron, Ronen Basri, and Irad Yavneh. Learning algebraic multigrid using graph neural networks. In *International Conference on Machine Learning (ICML)*, 2020. URL https://api.semanticscholar.org/CorpusID:212675750.

Jan Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3):233–241, 1993. doi: 10.1002/cnm.1640090307. URL https://api.semanticscholar.org/CorpusID:9525327.

Hugo Melchers, Victorita Dolean, and Michael Abdelmalik. When can a neural operator replace a coarse solve? architectural principles for two-level preconditioning, 2026. URL https://arxiv.org/abs/2605.19867.

Roy A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problems. *SIAM Journal on Numerical Analysis*, 24(2):355–365, 1987. doi: 10.1137/0724027. URL https://api.semanticscholar.org/CorpusID:121660604.

Luke N. Olson, Jacob B. Schroder, and Raymond S. Tuminaro. A new perspective on strength measures in algebraic multigrid. *Numerical Linear Algebra with Applications*, 17, 2010. URL https://api.semanticscholar.org/CorpusID:7590469.

Open Porous Media Initiative. opm-data: Open datasets for reservoir simulation. GitHub repository, revision eaa2261683a97027e057c2bc49612ad1c86390b3, 2026. URL https://github.com/OPM/opm-data/tree/eaa2261683a97027e057c2bc49612ad1c86390b3. Accessed 2026-09-05.

David Pfau, Stig Petersen, Ashish Agarwal, David G. T. Barrett, and Kimberly L. Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. In *International Conference on Learning Representations*, 2018. URL https://api.semanticscholar.org/CorpusID:86393936.

John W Ruge and Klaus Stüben. Algebraic multigrid. In *Multigrid Methods*, pp. 73–130. SIAM, 1987. doi: 10.1137/1.9781611971057.ch4. URL https://api.semanticscholar.org/CorpusID:124864296.

Nicole Spillane, Victorita Dolean, Patrice Hauret, Frédéric Nataf, Clemens Pechstein, and Robert Scheichl. Abstract robust coarse spaces for systems of pdes via generalized eigenproblems in the overlaps. *Numerische Mathematik*, 126:741 – 770, 2013. URL https://api.semanticscholar.org/CorpusID:253738013.

Jok Tang, Reinhard Nabben, Cornelis Vuik, and Yogi A. Erlangga. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. *Journal of Scientific Computing*, 39:340–370, 2009. URL https://api.semanticscholar.org/CorpusID:12770654.

Ulrich Trottenberg, Cornelis W Oosterlee, and Anton Schuller. *Multigrid*. Academic Press, 2000.

Zeyu Xiong, Ran Ma, Teng-fong Wong, and WaiChing Sun. Digital rock physics: Permeability and formation factor, 2024.

Jinchao Xu and Ludmil Zikatanov. Algebraic multigrid methods. *Acta Numerica*, 26:591–721, 2017. doi: 10.1017/S0962492917000083.

A METHOD DETAILS

The 2D and 3D domains are $[0,1]^2$ and $[0,1]^3$, discretized on uniform cell-centered grids with homogeneous Dirichlet boundaries. Axial transmissibilities use harmonic face averages. In the 3D benchmark, each boundary face contributes κ/h^2 to the adjacent diagonal, the one-sided cell-centered convention implemented throughout this study, rather than the $2\kappa/h^2$ half-cell flux convention. Every compared method uses the same operator. The 2D full-tensor operator adds the compact mixed-derivative stencil for the off-diagonal tensor entry; the 3D obstacle operator is scalar and uses the corresponding seven-point stencil. Independent tests verify symmetry, positive definiteness, exact reduction of the tensor assembly to the scalar assembly in the isotropic limit, analytic grid convergence for scalar diffusion, and the expected non-M-matrix structure for rotated tensors. The preconditioned CG recurrence uses a zero initial guess and stops when its recursive relative residual is at most 10^{-8}. Numerical result audits then recompute the true relative residual, which is below 10^{-8} for every audited solve unless a separate gate is stated. Timing-only records retain recursive convergence but not solution vectors. Synthetic right-hand sides are deterministic Gaussian random fields; all methods within a comparison receive identical vectors.

Table 3 summarizes the two controlled benchmarks. Solver definitions follow the table.

Table 3: The controlled benchmarks isolate tensor anisotropy in 2D and topological barriers in 3D. Benchmark and training specification.

Component	2D rotated anisotropy	3D obstacle topology
Coefficient	Rotated tensor	Barrier field, contrast 10^4
Training grid	256^2	48^3
Learned rank	32	16
Training operators	200	64 or 200
Training epochs	80	250
Network inputs	Tensor, coordinates	Conductivity, coordinates

The 2D tensor angle is uniform on $[16, 25]$ degrees and its anisotropy ratio is uniform on $[200, 5000]$. The 3D generator places three to six low-conductivity barrier planes in the central three quarters of the final axis, each with one random circular opening. Noncoordinate channels are standardized using training-set statistics; coordinates are not normalized. Training uses Adam with batch size one, initial learning rate 10^{-3}, cosine decay, and gradient-norm clipping at one.

Algebraic multigrid (AMG) constructs its hierarchy directly from the sparse matrix. Our PyAMG baseline uses Ruge–Stüben coarsening, classical strength with threshold 0.25, classical interpolation, and symmetric Gauss–Seidel smoothing; the stronger variant changes only the strength measure to evolution strength. BoomerAMG is HYPRE's parallel AMG implementation, accessed through PETSc. It uses one V-cycle, symmetric hybrid symmetric Gauss–Seidel (SOR/Jacobi) smoothing, and an exact coarse solve in our tests. Geometric multigrid (GMG) instead uses the known grid hierarchy. Our 3D implementation, GMG3D, uses trilinear prolongation, Galerkin coarsening, weighted-Jacobi smoothing, and a sparse LU coarse solve. Unless stated otherwise, $n_{\min} = 8$ and two smoothing sweeps are used.

All corrected methods use Eq. (2). Coarse arrays and the regularized Cholesky factor use single precision, with diagonal shift $10^{-12}\,\mathrm{tr}(E)/m$; outer vectors and the base preconditioner use double precision.

A.1 Suites, checkpoints, and reporting

Table 4 collects the terms used throughout the paper.

Table 4: Terms used throughout the paper.

Term	Meaning
Base preconditioner M_1	The deployed multigrid V-cycle ("one cycle", uncorrected); "two cycles" applies it twice
Solver alignment	Choosing the coarse space from the error M_1 leaves behind, not from $\mathbf{A}$ alone
Coarse space $\mathbf{Z}$; correction	The rank-m span; the preconditioner M_c of Eq. (2) built from it
Exact Rayleigh space	Exact minimizer of the operator-only Rayleigh objective, Eq. (3)
Ideal solver space	Bottom-m invariant subspace of H_1, i.e. the slowest directions of $I - M_1\mathbf{A}$ (Lemma 1)
Finite solver target	Its finite approximation $\widehat{Z}$, the training label of Eq. (5)
Predicted space	Network output after the deployment map $\mathrm{qr}(\mathbf{D}^{-1/2}V)$
Rayleigh / solver-aligned FNO	The same network trained with $\mathcal{L}_{\mathrm{RQ}}$ / $\mathcal{L}_{\mathrm{SA}}$
Target reuse	Deploying the nearest stored training target, or an interpolation of four, instead of a prediction
Suite	An independently generated set of held-out operators for one comparison (App. A.1)

Table 5 summarizes where each of the three requirements holds, with the section giving the evidence.

Table 5: Where each requirement holds. *Target*: a solver-aligned target improves the base preconditioner. *Prediction*: a network recovers that benefit on new operators. *Cost*: the learned correction beats strong classical solvers in total time. "–" means not tested.

Setting	Target	Prediction	Cost
2D, inside the training band (§4.2–4.4, 4.7)	Yes	Yes	No
2D, outside the training band (§4.4)	–	No	–
2D, lognormal coefficients (§4.4)	Yes	No	–
3D obstacles (§4.5)	Yes	No	–
3D sandstone scans (§4.5)	Mixed	–	–
3D transfer to new geometries (§4.6)	–	No	–

Table 6 lists every held-out suite named in the main text. Each suite is generated independently, and rows from different suites are never pooled.

Checkpoints. The original 2D Rayleigh checkpoint also received a generated source field that is absent from the loss and solver; a retrained checkpoint without that channel gives identical Evolution AMG counts on all 12 Baseline operators and is used for the BoomerAMG row of Table 1. The rank-64 Bentheimer study uses a separate 3D checkpoint of the same architecture trained for 250 epochs on 64 synthetic granular fields of random packed low-conductivity spheres. Records keep the historical name `solver_informed` for the solver-aligned objective and its models.

2D controls. Rotated-anisotropy controls include three training seeds, stronger PyAMG variants, PETSc/HYPRE BoomerAMG, four fixed GenEO budgets, adaptive GenEO, geometric spaces, a training-snapshot POD basis, independent operator assembly, and resolution scaling (Appendix C.2).

Reporting. Let k_{jr} be the iteration count for operator j and right-hand side r and k_j its mean over right-hand sides. Equation (6) averages per-operator reductions; a caption says explicitly when a reduction in aggregate means is reported instead. Sample standard deviations and descriptive intervals describe the recorded finite sets; they do not license population inference or adjust for

Table 6: Held-out suites. Operators are the independent units; right-hand sides (RHS) are repeated measurements.

Suite	Question	Requirement	Operators	RHS	Section
Baseline	Stronger base preconditioners	Setup	12	1 or 16	4.1
Objective	Rayleigh vs. solver target	Target	24	8	4.2
Crossed	Target moves with M_1	Target	24	8	4.2
Architecture	FNO vs. U-Net	Target	24	8	4.3
Retrieval	Prediction vs. reuse	Prediction	24	8	4.4
Disjoint-angle	Extrapolation	Prediction	24	8	4.4
Lognormal validation	Second 2D family	Prediction	12	8	4.4
3D development	3D target and prediction	Both	6	8	4.5
3D resolution	Transfer across grids	Prediction	3 per size	8	4.6
Public bone volumes	Transfer across topology	Prediction	8	8	4.6
Timing	Time vs. cycles and reuse	Cost	24	up to 64	4.7
Common-hardware	Time vs. BoomerAMG	Cost	24	up to 64	4.7

multiple comparisons. The principal 2D Baseline table uses one deterministic right-hand side per operator; batched 3D results keep separate stopping counts for every right-hand side.

A.2 Exact solver alignment

A.2.1 A counterexample for solver ordering

The normalized Rayleigh objective need not select the space preferred by the base preconditioner. Fix $0 < \rho < 1$ and $0 < \epsilon < 1$, and define

$$q_- = 2^{-1/2}(1,-1)^\top, \quad q_+ = 2^{-1/2}(1,1)^\top, \quad \lambda_- = 1-\rho, \quad \lambda_+ = 1+\rho, \tag{7}$$

$$\mathbf{A} = [q_-\ q_+]\,\mathrm{diag}(\lambda_-,\lambda_+)[q_-\ q_+]^\top, \qquad M_1 = [q_-\ q_+]\,\mathrm{diag}(\lambda_-^{-1},\epsilon\lambda_+^{-1})[q_-\ q_+]^\top. \tag{8}$$

Both matrices are SPD and $\mathrm{diag}(\mathbf{A}) = I$. Consequently, q_- spans the unique rank-one minimizing subspace of Eq. (3). For either eigenvector $z = q_j$, Eq. (2) sets the selected eigenvalue of $M_c\mathbf{A}$ to one and leaves the unselected eigenvalue of $M_1\mathbf{A}$ unchanged. Since $M_1\mathbf{A}$ has eigenvalues $(1,\epsilon)$, choosing $z = q_-$ gives spectrum $(1,\epsilon)$ and condition number $1/\epsilon$. Choosing the higher-Rayleigh vector $z = q_+$ gives spectrum $(1,1)$ and condition number one. The separation can therefore be arbitrarily large even with exact optimization and a fixed SPD base preconditioner. This result concerns spectral conditioning, not an arbitrarily large iteration gap in two dimensions.

The exact minimizer of Eq. (3) is the bottom-m eigenspace of $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$. We compute a rank-64 basis with AMG-preconditioned LOBPCG and use prefixes at ranks 16 and 32. The largest final eigenresidual is below 10^{-7} across the three Bentheimer operators.

There is a precise limit to what the scalar trace certifies. For $1 \le m < n$, let $H = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ have eigenvalues $\lambda_1 \le \cdots \le \lambda_n$, let U span its bottom m eigenvectors, and define $L(Q) = m^{-1}\,\mathrm{tr}(Q^\top HQ)$ for orthonormal Q. Then

$$L(Q) - L(U) \ge \frac{\lambda_{m+1}-\lambda_m}{m}\,\|\sin\Theta(U,Q)\|_F^2. \tag{9}$$

To see this, expand the projector $QQ^\top$ in the eigenbasis of H and pair its total mass outside the bottom eigenspace with the same missing mass inside it. In particular, replacing the mth eigenvector by the $(m+1)$st changes the mean trace by only $(\lambda_{m+1}-\lambda_m)/m$ while making the minimum principal cosine zero. A small boundary gap therefore makes trace loss weakly informative about coverage of individual boundary modes. This bound does not say whether such a mode matters to a chosen multigrid cycle; that requires the matched solver comparisons in Section 4.2.

We then apply the same deployment transformation as the network:

$$\mathbf{Z}_{\mathrm{RQ}} = \mathrm{qr}(\mathbf{D}^{-1/2}\mathbf{Q}_{\mathrm{bottom}}). \tag{10}$$

The finite solver-target comparison iterates $Q \leftarrow \mathrm{qr}(Q - M_{\mathrm{GMG}}\mathbf{A}Q)$ for 15, 25, or 40 steps, targeting slowly damped modes of the fixed GMG3D cycle. Both spaces are inserted into Eq. (2) and evaluated on the same 32 right-hand sides with per-right-hand-side stopping counts.

A.3 PROSPECTIVE CROSS-ARCHITECTURE CONTROLS

The solver-aligned control changes the model family more substantially. A three-level convolutional U-Net has 2.44M parameters, within 1% of the 2.41M-parameter FNO. Both models use the same 200 training operators, serialized rank-32 targets, overlap loss, 80-epoch optimizer schedule, diagonal scaling, QR map, and Evolution AMG configuration. A 12-operator validation gate passed before the seed-1713 test was evaluated. On that test, FNO is lower on 19 operators and ties on five; its paired advantage over U-Net is 0.82 iterations with descriptive interval $[0.55, 1.09]$.

After these three architecture-objective cells were known, we trained the missing Rayleigh U-Net with the same 200 operators, 80 epochs, and initialization seed as the matched Rayleigh FNO. This is a retrospective completion, not a prospectively selected model comparison. Before generating any evaluation operator, we fixed a new seed-1744 stream of 24 operators, eight right-hand sides per operator, checkpoint and source hashes, and unconditional reporting. On that stream, solver-aligned training lowers iterations on all 24 operators for both architectures. The paired Rayleigh-minus-solver-aligned differences are 3.91 $[3.18, 4.63]$ for FNO and 3.47 $[2.71, 4.23]$ for U-Net. One Evolution AMG cycle averages 28.11 iterations and two cycles average 19.27. All 1,152 solves have recomputed true relative residual below 10^{-8}. Table 2 reports the full matched matrix.

B NUMERICAL VALIDATION

We test matrix symmetry and positive definiteness, scalar and tensor assembly, PCG convergence, multigrid cycles, the balancing correction, and the GenEO eigenproblem on independently constructed small problems. Central reported relationships are also recomputed from retained per-instance records rather than from rounded table entries. Checkpoints are identified by cryptographic hashes, while large eigenspace caches are regenerated from the released inputs.

We treat the retrieval and early 3D studies as retrospective because their records do not establish prospective status. This qualification does not change their solver outcomes or the ordering on the prospectively held-out 2D test.

B.1 PROOF OF THE SPECTRAL CHARACTERIZATION

For SPD $\mathbf{A}$ and M_1 and full-rank $\mathbf{Z}$, define

$$W = \mathbf{A}^{1/2}\mathbf{Z}(\mathbf{Z}^\top\mathbf{A}\mathbf{Z})^{-1/2}, \quad P = WW^\top, \quad H_1 = \mathbf{A}^{1/2}M_1\mathbf{A}^{1/2}. \tag{11}$$

Then $W^\top W = I$, and transforming Eq. (2) gives

$$\mathbf{A}^{1/2}M_{\mathrm{c}}\mathbf{A}^{1/2} = P + (I-P)H_1(I-P). \tag{12}$$

Suppose $1 \le m < n$ and H_1 has eigenvalues $0 < \mu_1 \le \cdots \le \mu_n \le 1$. The condition-number-optimal transformed rank-m space can be its bottom-m eigenspace, and the optimal condition number is $1/\mu_{m+1}$. Equation (12) has eigenvalue one on $\mathrm{range}(W)$ and the spectrum of the compression of H_1 to its orthogonal complement elsewhere. The Courant and Fischer variational principle bounds the smallest compressed eigenvalue by μ_{m+1} for every $(n-m)$-dimensional complement. The bottom-m eigenspace attains this bound; degenerate spectra can admit other optimizers.

For completeness, the contraction premise holds in exact arithmetic for the released PyAMG Ruge–Stüben V-cycle. Let $E_\ell = I - M_\ell A_\ell$ be the error map at level ℓ. Symmetric Gauss–Seidel has error map $S_\ell = F_\ell^{*_{A_\ell}} F_\ell$, where F_ℓ is a forward sweep and $*_{A_\ell}$ denotes the adjoint in the A_ℓ inner product. Thus S_ℓ is an A_ℓ-self-adjoint positive-semidefinite contraction. PyAMG uses $R_\ell = P_\ell^\top$ and $A_{\ell+1} = P_\ell^\top A_\ell P_\ell$. Define $Q_\ell = A_\ell^{1/2}P_\ell A_{\ell+1}^{-1/2}$, for which $Q_\ell^\top Q_\ell = I$. If the transformed $E_{\ell+1}$ is a positive-semidefinite contraction, the transformed inexact coarse-correction error is $\widehat{C}_\ell = I - Q_\ell(I - \widehat{E}_{\ell+1})Q_\ell^\top$, also a positive-semidefinite contraction. Pre- and post-smoothing then give the adjoint sandwich $E_\ell = S_\ell^{*_{A_\ell}} C_\ell S_\ell$, which preserves both properties. The terminal exact solve has zero error, so induction proves $0 \preceq_{A_0} E_0 \preceq_{A_0} I$. Since $A_0^{1/2}E_0A_0^{-1/2} = I - H_1$, this gives $0 \prec H_1 \preceq I$ when the resulting V-cycle preconditioner is positive definite. Direct small-matrix checks of classical and evolution strength confirm the hierarchy identities, symmetry, and generalized spectral bounds. This implementation-specific argument does not cover HYPRE

BoomerAMG, whose relaxation path differs. The same argument covers GMG3D: weighted Jacobi with $\omega_\ell = (4/3)/\rho(D_\ell^{-1}A_\ell)$ has an A_ℓ-self-adjoint error map with spectrum in $[-1/3, 1]$ when ρ is exact, and an energy contraction whenever the power-iteration estimate is at least $2\rho/3$, GMG3D applies the same number of sweeps before and after the Galerkin coarse correction, and the coarsest solve is exact, so the V-cycle error map is again an A-self-adjoint positive-semidefinite contraction.

Finite block iteration approaches the stated bottom eigenspace only conditionally: the target block must overlap that eigenspace, the rank boundary must be separated, and the relevant error-propagation spectrum must order the desired modes. Moreover, the network's Euclidean overlap loss does not have a proved bound on the transformed energy-angle δ below. The theorem thus characterizes an exact target; it does not certify the finite target, learned space, or regularized mixed-precision deployment.

The characterization also gives a finite-approximation guarantee under the same assumptions. Let U_* span a bottom-m eigenspace of H_1, and define the worst transformed-space error $\delta = \|(I-P)U_*\|_2$. For any unit $x \in \mathrm{range}(W)^\perp$, the bottom-space component satisfies $\|U_*^\top x\|_2 \le \delta$, so $x^\top H_1 x \ge \mu_{m+1} - (\mu_{m+1} - \mu_1)\delta^2$. Equation (12) has eigenvalue one on $\mathrm{range}(W)$ and complementary eigenvalues at most one. Consequently, the exact-arithmetic bound below recovers $1/\mu_{m+1}$ at $\delta = 0$ and controls sufficient degradation by the largest principal angle; it is not an iteration or runtime guarantee:

$$\kappa_2(\mathbf{A}^{1/2} M_{\mathrm{c}} \mathbf{A}^{1/2}) \le \frac{1}{\mu_{m+1} - (\mu_{m+1} - \mu_1)\delta^2}. \tag{13}$$

B.2 Solver-aligned prospective protocol

The solver-aligned extension was frozen after the exact characterization and a three-operator development screen, but before accessing generator seeds 1701 and 1702. For each of the same 200 training operators used by the Rayleigh FNO, we initialize a rank-32 Gaussian basis with seed $61000 + i$ and apply 64 steps of $I - M_1 A$, reorthonormalizing after each step. Here M_1 is one symmetric Evolution AMG cycle with the same settings later used at evaluation. The historical producer seeded the Gaussian block but not hierarchy construction, so the released targets are checksum-fixed realizations rather than bitwise-regenerable outputs of one identified M_1. The independent post-serialization audit constructed another unseeded hierarchy; its residual therefore cannot be attributed solely to serialization. The serialized float32 targets have maximum orthogonality error 6.56×10^{-7} and maximum relative invariance residual 0.0595.

The FNO starts from random initialization and minimizes Eq. (5) for 80 epochs. Its final mean one-minus-overlap is 0.0946. Architecture, 2.41 million parameters, operator inputs, diagonal map, optimizer, schedule, rank, and deployment correction match the Rayleigh model. Only the training signal differs. Generating the 200 labels requires 64 rank-32 block iterations per operator: 409,600 single-vector V-cycle applications in total. The generator and post-serialization invariance checks add 12,800 applications. These costs are outside the 739-second network-training time and outside deployment.

The 12-operator validation gate required improvement on at least 10 operators, no loss larger than two percent of the uncorrected count relative to Rayleigh, at least a 25% aggregate cut, and at least a five-percentage-point advantage over Rayleigh. The solver-aligned FNO passed all four conditions: it improved all 12, cut 31.42%, and exceeded Rayleigh by 14.06 percentage points. Only then was the 24-operator test stream generated. Figure 2a reports that unconditionally retained outcome. All 1,152 test solves converge with recomputed true relative residual below 10^{-8}. Operators are the independent units; the eight right-hand sides are repeated measurements.

After retaining that test, we evaluated a second holdout at generator seed 1703 to separate learned interpolation from direct reuse of the training targets. We first trained two additional random initializations without validation or test access. Their final one-minus-overlap losses are 0.0938 and 0.0945, compared with 0.0946 for the original initialization. The holdout compares all three networks with the nearest training target and a four-neighbor interpolation. Distance is Euclidean after mapping angle and log anisotropy to the unit square. The interpolation weights are inverse squared distances; its rank-32 space is the leading left singular subspace of the four weighted target bases.

Table 7 reports every outcome. The FNO mean is lower than nearest reuse on 14 operators, tied on 9, and higher on 1. It is lower than subspace interpolation on 11, tied on 12, and higher on 1. The paired intervals exclude zero, while the comparison with two AMG cycles does not. This supports stable amortization only inside the controlled training family.

Table 7: Prediction modestly improves on direct target reuse within the controlled family. Retained seed-1703 retrieval control over 24 operators and eight right-hand sides each. FNO reports the mean over three independently initialized networks. Differences are comparator minus FNO, so positive values favor FNO.

Method	Iterations	Difference	Descriptive 95% interval
Evolution AMG, one cycle	29.20		
Evolution AMG, two cycles	19.97	-0.210	$[-0.702, 0.282]$
Nearest training target	20.62	0.436	$[0.203, 0.668]$
Four-neighbor target interpolation	20.46	0.280	$[0.083, 0.476]$
Solver-aligned FNO	20.18		

A quiet-host storage audit separates prediction from target reuse beyond iteration counts. Table 8 gives exact stored-byte ratios relative to the 18.2 MiB FNO checkpoint and warm page-cache loading times. Generic lossless compression reduces float32 targets only modestly, whereas float16 halves their size. Reorthonormalizing float16 targets preserves all 192 iteration counts on the 24-operator retrieval suite. Thus prediction has a measured storage advantage, but lookup is already below 0.6 ms and the end-to-end timing comparison with reuse remains unresolved.

Table 8: The learned checkpoint is far smaller than a reusable target library. Target-library storage and warm loading; stored sizes are ratios to the 18.2 MiB FNO checkpoint; loading materializes each float32 prefix.

Targets	Float32	Float16	Compressed float32	Load (ms)
8	3.51×	1.76×	3.29×	10.23
32	14.04×	7.02×	13.17×	37.78
64	28.09×	14.04×	26.33×	74.11
200	87.77×	43.89×	82.32×	228.42

A third protocol was frozen before accessing generator seed 1706. It compares online block iteration of the error propagator of the base preconditioner with the fixed FNO on 24 new operators. Each block iteration costs 32 first-level V-cycles. The online construction improves steadily but does not match the FNO at any prespecified budget through 64 steps (Table 9). Two AMG cycles and FNO are numerically tied in aggregate. This establishes amortization in outer iterations only; the comparison is not a timing result.

Table 10 reports a fourth frozen holdout crossing rank-32, 64-step targets generated by default and Evolution AMG with deployment around either preconditioner.

Matched minus crossed is -29.39 iterations under default AMG and -2.94 under Evolution AMG, with descriptive intervals $[-32.13, -26.65]$ and $[-3.38, -2.49]$; the matched target wins all 24 operators for both preconditioners. This isolates preconditioner-specific target selection within one family, not neural transfer or matched computational work.

A fifth protocol freezes 24 new operators, 64 right-hand sides, three hierarchy seeds, and cyclic method order before timing. Every method rebuilds the same seeded Evolution AMG hierarchy within a repeat. Per-operator setup includes the fresh hierarchy and, where applicable, inference, diagonal scaling, QR, retrieval or interpolation, and balancing-correction construction. One-time checkpoint and target-database loading are excluded. This is warm-loaded per-operator deployment timing, not process-start timing. Table 11 reports equal-operator mean total time.

For one right-hand side, one-cycle AMG is 0.074 seconds faster than FNO, with paired descriptive interval $[0.027, 0.121]$. At eight right-hand sides, FNO is 0.233 seconds faster, with interval $[0.108, 0.358]$. At 64, FNO is 15.2% faster than one cycle and 37.0% faster than two cycles. Differ-

Table 9: Amortized prediction matches 64-step online construction without its per-operator setup work. Frozen seed-1706 frontier over 24 operators and eight right-hand sides each. Setup work counts rank-32 V-cycle applications per operator. Solve-time hierarchy work and all correction costs remain additional.

Method	Setup V-cycles	Iterations
Evolution AMG, one cycle	0	29.49
Online target, 1 block step	32	29.48
Online target, 4 block steps	128	27.70
Online target, 8 block steps	256	25.58
Online target, 16 block steps	512	23.29
Online target, 32 block steps	1,024	21.23
Nearest training target	0 online	21.04
Interpolated training target	0 online	20.85
Online target, 64 block steps	2,048	20.31
Evolution AMG, two cycles	0	20.21
Solver-aligned FNO	0 online	20.22

Table 10: Targets generated by the base preconditioner win every crossed comparison. Frozen study over 24 operators and eight right-hand sides each. Targets share one initial space and use 64 error-propagation block iterations. Lower iteration counts are better. The final column is matched minus crossed with an operator-level descriptive interval.

Deployment	One cycle	Two cycles	Matched	Crossed	Difference [95% interval]
Default AMG	105.70	72.92	69.11	98.51	$-29.39\ [-32.13, -26.65]$
Evolution AMG	27.10	18.60	18.91	21.85	$-2.94\ [-3.38, -2.49]$

ences from nearest reuse and interpolation are not separated in this record at 64. This is a single-host Python and SciPy measurement on the narrow training family, not a production, cold-start, or cross-family speed claim. Eight is the first prespecified reporting-grid crossing, not an estimate of the exact break-even point.

A separately frozen common-hardware audit compares the FNO with both default BoomerAMG and the development-selected strong threshold 0.5. It uses 24 new seed-1711 operators, 64 right-hand sides, three cyclic method-order repeats, and the same single host. Table 12 includes fresh hierarchy construction and method-specific setup. Both BoomerAMG variants are substantially faster at every budget, so iteration reductions against the matched Evolution AMG preconditioner do not imply superiority to a strong production baseline. Offline target generation and training remain additional costs.

B.3 Cost accounting

Table 13 collects the 64-right-hand-side totals used in Section 4.7, and Figure 5 shows every solve budget. Table 14 gives a per-iteration flop model for the 24 Common-hardware operators, computed after the timing study from the recorded hierarchies (post hoc, no training and no solves). A V-cycle costs $\sum_\ell (2s+2)\,\mathrm{nnz}(A_\ell) + 4\,\mathrm{nnz}(P_\ell)$ over non-coarsest levels plus $2\,\mathrm{nnz}$ of the coarsest matrix, where $s = 4$ directional smoothing sweeps per cycle for both PyAMG's symmetric Gauss–Seidel and HYPRE's hybrid symmetric Gauss–Seidel; the rank-32 correction costs $8mN$, and a matrix-vector product $2\,\mathrm{nnz}(\mathbf{A})$. Vector operations of CG are omitted. Iterations are the recorded means of the timing study. The model ignores memory traffic and implementation, which is why measured time can order methods differently.

B.4 Spectrum diagnostics

Figure 2c and Figure 6 use the 64 largest eigenvalues μ of the error propagator $I - M\mathbf{A}$, computed by a Lanczos solver on the symmetric generalized problem $\mathbf{A}(I - M\mathbf{A})x = \mu\mathbf{A}x$ with tolerance 10^{-8}, and report $\lambda = 1 - \mu$. The correction uses double-precision coarse arrays. Both studies are

Table 11: Within the controlled family, FNO setup amortizes after several right-hand sides but does not separate from target reuse. Frozen seed-1707 timing in seconds per operator. Each entry includes setup and the indicated cumulative solve budget, averaged first over three paired stochastic-hierarchy repeats and then over 24 operators.

Method	$R = 1$	$R = 8$	$R = 32$	$R = 64$
Evolution AMG, one cycle	0.532	2.349	8.559	16.805
Evolution AMG, two cycles	0.623	3.061	11.471	22.623
Nearest training target	0.528	2.060	7.335	14.357
Four-neighbor target interpolation	0.675	2.197	7.420	14.388
Solver-aligned FNO	0.606	2.117	7.326	14.255

Table 12: Both BoomerAMG configurations are faster than learned deployment at every solve budget. Common-hardware total time in seconds per operator; means first average three cyclic-order repeats, then 24 operators.

Method	$R = 1$	$R = 8$	$R = 32$	$R = 64$
Default BoomerAMG	0.148	0.890	3.432	6.820
Threshold-0.5 BoomerAMG	0.155	0.959	3.719	7.390
Evolution AMG	0.547	2.519	9.320	18.352
Solver-aligned FNO	0.660	2.569	9.107	17.847

post hoc and descriptive on the single Figure 1 operator; their protocols were frozen and committed before the runs. The first protocol named, as its count summary, the number of eigenvalues below 0.1 among the 64 smallest; that count is zero in every case, so the main text reports the smallest eigenvalue instead. The interlacing implied by Eq. (4) holds for every space.

The second study builds rank-32 finite solver targets with default AMG and with Evolution AMG from one shared initial block and 64 block iterations, and deploys each around both preconditioners (Figure 6). Around Evolution AMG, the matched target nearly reaches the ideal solver space (smallest eigenvalue 0.217 against 0.223; this target uses a different random initial block than the one in Figure 2c, which gives 0.210), while the crossed default-AMG target (0.134) is below even the exact Rayleigh space (0.154). Around default AMG the picture reverses: the exact Rayleigh space is almost the ideal solver space (0.0257 against 0.0261). Both finite targets miss the lowest cluster of eigenvalues, the matched one less (0.0114 against 0.0066 for the crossed target, equal to default AMG alone). The ideal space therefore moves with the preconditioner: around default AMG the slowest directions are close to the operator's low-energy modes, which is consistent with the large Rayleigh gain against default AMG in Table 1, and around Evolution AMG they are not.

B.5 Stopped 3D solver-aligned development

We completed the exposed six-operator development matrix before deciding whether to open a reserved confirmation stream. Three compact rank-16 FNOs have 1.33 million parameters and use the same 64 training operators, 250 epochs, optimizer, and deployment map. They differ only in Rayleigh, Euclidean finite-target, or energy-norm loss. The table also retains the earlier 37.8-million-parameter checkpoints for context. The finite target is effective, but none of the five predictors passes the frozen gate, which required a Compact candidate to beat the Large solver-aligned FNO on at least five of the six operators and by at least 0.5 iterations on average, with no operator more than 2% worse than one cycle; no confirmation stream is accessed.

For the energy-norm arm, let U_A and $\widehat{U}_A$ be A-orthonormal bases for the predicted and target spaces. Its loss is the mean squared sine of their energy-space canonical angles, $\mathcal{L}_A = 1 - m^{-1}\|\widehat{U}_A^\top A U_A\|_F^2$.

The Large solver-aligned FNO attains mean squared Euclidean subspace error 0.4714 and energy-space error 0.8117 on development operators. The Compact solver-aligned FNO has corresponding errors 0.6217 and 0.9329, so reducing capacity does not regularize the relevant geometry in this test.

Table 13: Only the classical-baseline comparison decides cost. Total time in seconds per operator at 64 right-hand sides, including all setup. The two suites use different operators and are not paired; compare rows only within a block. Appendix B.2 gives all solve budgets and intervals.

Suite	Method	Time (s)
Timing	Evolution AMG, two cycles	22.62
	Evolution AMG, one cycle	16.81
	Nearest training target	14.36
	Solver-aligned FNO	14.26
Common-hardware	Evolution AMG, one cycle	18.35
	Solver-aligned FNO	17.85
	Threshold-0.5 BoomerAMG	7.39
	Default BoomerAMG	6.82

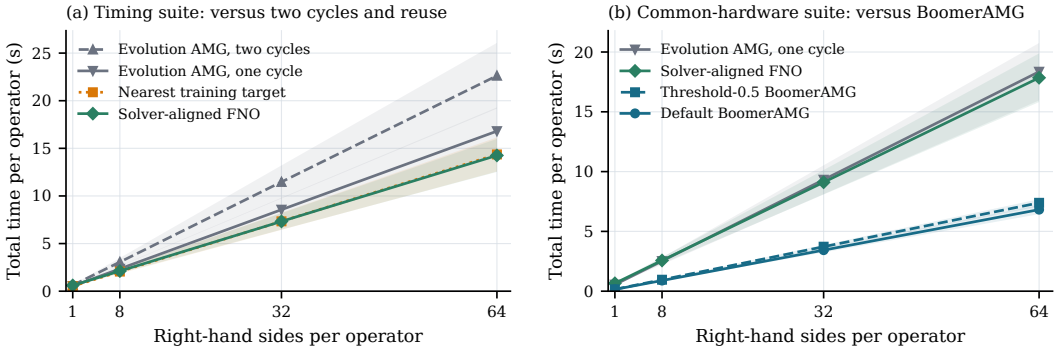

Figure 5: Total time per operator against the number of right-hand sides, with descriptive 95% bands over 24 operators. The panels use different, unpaired suites and should not be compared with each other.

Two alternate target initializations remain effective at 14.38 and 14.29 iterations, compared with 14.33 for the reference, so target instability does not explain the 18.81 prediction mean. At fixed rank, replacing the one, two, or four worst energy-principal directions lowers the mean to 18.52, 17.81, and 17.15. Four random four-direction controls average 18.05. Thus a few missed solver-relevant directions explain part, but not all, of the gap. These are development diagnostics on one family and do not rule out other architectures, data scales, or worst-angle objectives.

B.6 Hierarchy-realization sensitivity

The crossed-target conclusion is unchanged across three independently constructed PyAMG hierarchies. We audit eight exposed operators with matched right-hand sides and fixed target-construction blocks. Table 16 reports matched minus crossed iterations, so negative values favor the target generated by the deployed preconditioner. This audit tests realization stability, not population-level generality.

B.7 Disjoint-angle generalization

We froze a second 2D suite before evaluation to test angles in $[27, 36]^\circ$, disjoint from the training interval $[16, 25]^\circ$. It uses 24 operators, eight shared right-hand sides, the released checkpoint, and the same nearest-target and four-neighbor retrieval rules as the in-band study. Table 17 shows that prediction remains slightly better than one Evolution AMG cycle but loses to both retrieval controls and a second cycle. The prespecified separation rule fails. The positive 2D learned result therefore supports interpolation within the controlled family, not angular extrapolation.

Table 14: The correction costs more per iteration than a V-cycle. Flop model per unknown, averaged over the 24 Common-hardware operators; "Grid" and "Operator" are complexities. "Per solve" multiplies the per-iteration cost by the recorded mean iterations.

Method	Levels	Grid	Operator	Per iter.	Iter.	Per solve
Evolution AMG, one cycle	9.2	1.77	2.05	$214N$	28.44	$6.1\text{k}N$
Evolution AMG + correction				$470N$	19.49	$9.2\text{k}N$
Evolution AMG, two cycles				$411N$	–	–
Default BoomerAMG	10.0	1.79	2.12	$226N$	10.34	$2.3\text{k}N$
Threshold-0.5 BoomerAMG	12.1	1.89	2.21	$234N$	10.63	$2.5\text{k}N$

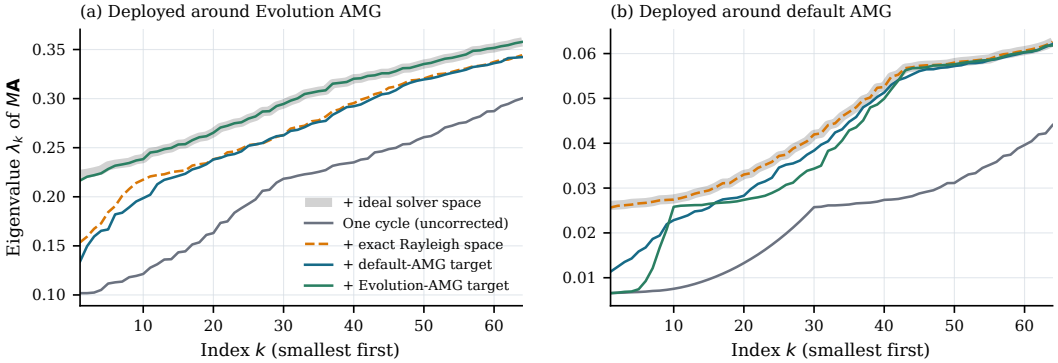

Figure 6: The ideal space moves with the base preconditioner. The 64 smallest eigenvalues of the preconditioned operator on the Figure 1 operator, rank 32; higher is better. Finite solver targets are built with default AMG or Evolution AMG from one shared block and deployed around each preconditioner.

B.8 Negative and screened-out results

Every study run for the solver-aligned objective is reported here, including those that stopped. Each family first passes a target screen: the exact finite solver target must improve the base preconditioner before any network is trained. Table 18 lists the outcome of each screen and, where a model was trained, of its validation. Records, protocols, and validators are in the supplement; the checkpoints of these studies were not retained.

Protocol chronology. Protocol freeze times were typed by hand. For 16 protocols the recorded time is later than the protocol's first commit, and for 13 of them a bound record is timestamped before the recorded freeze. In every case the protocol's first commit precedes every timestamped record that binds it. The supplement contains the complete audit and the tool that repeats it; the audit modifies no protocol or record.

C Rotated-anisotropy details

Across the 12 held-out operators, the default-AMG percentage cut has sample standard deviation 1.55 percentage points. With Evolution AMG, the mean paired improvement is six iterations and the mean per-operator iteration reduction is 18.0%, with descriptive interval $[16.1, 20.0]\%$. The corresponding BoomerAMG interval and threshold-selection study are reported below.

C.1 Classical and learned frontier

The adaptive rule keeps each local generalized eigenvector whose generalized eigenvalue exceeds τ. The thresholds are an empirical sweep of the selection rule; they are not presented as reproducing a theoretical condition-number constant for this partition. Candidate pools are doubled until each local spectrum includes a below-threshold mode, ruling out silent truncation. The rank-32 fixed construction uses a 2×2 partition and eight modes per subdomain, whereas the adaptive construction

Table 15: Matched capacity and loss controls do not close the 3D target-to-prediction gap. Exposed development results over six 3D obstacle operators and eight right-hand sides each. Lower iterations are better; cut is the mean per-operator reduction (Eq. 6).

Method	Iterations	Cut
GMG3D, one cycle	24.75	0.00%
GMG3D, two cycles	17.40	29.71%
Finite solver target	14.33	41.56%
Large Rayleigh FNO	18.04	27.35%
Large solver-aligned FNO	18.81	23.68%
Compact Rayleigh FNO	22.48	10.08%
Compact solver-aligned FNO	21.56	12.59%
Compact solver-aligned FNO, energy loss	22.73	7.49%

Table 16: Preconditioner alignment is stable across the three audited hierarchy realizations. Means are over eight operators; matched wins count operators.

Base preconditioner	Realization 1	Realization 2	Realization 3	Matched wins
Default AMG	-31.13	-31.41	-31.47	$8/8$ each
Evolution AMG	-2.81	-2.84	-2.88	$8/8$ each

uses a 4×4 partition; both use two-cell overlap. The frontier therefore compares complete fixed and adaptive configurations, not a threshold-only ablation. Our implementation is also a repurposing of a GenEO-derived coarse space around AMG, not a complete reproduction of the original Schwarz method. In particular, the 1×1 fixed-rank construction solves one global eigenproblem and is a diagnostic reference rather than a scalable GenEO deployment. Verification tests cover the partition of unity, local Neumann operator, generalized eigenpairs, and solver improvement.

The strongest fixed-rank GenEO-derived construction wins seven operators and ties one, but three large losses reverse its aggregate mean. Its median paired difference from learned prediction is -0.47 iterations. Its iteration standard deviation is 8.57, compared with 6.99 for prediction. Thus the learned space is less variable on this suite, not uniformly better.

The paired fixed-rank control in Table 19 uses one universal POD basis: the leading 32 left singular vectors of 200 unit-norm training solutions. The basis is never conditioned on a test operator. Learned prediction is lower on all 12 paired operator means, by 4.18 iterations on average. A second frozen suite compares three GenEO-derived partitions with total rank 32, nearest-training retrieval, learned prediction, and the exact target Rayleigh space. Retrieval selects in normalized angle and log-anisotropy-ratio coordinates, computes the selected training operator's exact normalized bottom-32 eigenspace, and applies the target diagonal map. It uses no network. The normalized nearest-neighbor squared distance ranges from 3.7×10^{-7} to 8.0×10^{-3}, with median 5.1×10^{-4}. Across the 32 retained eigenvalues, the largest per-operator relative mismatch between retrieved and target spectra is 3.1%. The training set therefore samples this two-parameter family densely, which explains why lookup is a strong control rather than evidence of broad transfer. All 1,344 solves pass a recomputed relative-residual ceiling of 10^{-7}.

A causal input intervention cyclically replaces the three coefficient channels with those of another held-out operator while preserving target coordinates, operator, diagonal map, hierarchy, and right-hand sides. Correct inputs average 26.68 iterations versus 27.84 after permutation; correct is better on nine operators, tied on two, and worse on one. The paired mean difference is 1.16 iterations with descriptive interval $[-0.13, 2.44]$. This shift is suggestive of channel use, but the interval crosses zero and does not establish an effect at the current sample size. The retrieval comparison likewise leaves the need for network inference unresolved on this low-dimensional family. A separate arithmetic sensitivity check evaluates the same learned spaces and right-hand sides with the released single-precision coarse arrays and with the full balancing correction retained in double precision. The stopping count is identical for all 192 paired solves, and every recomputed residual is below 10^{-8}.

Table 17: The learned advantage does not extrapolate to the disjoint angle band. Equal-operator means over 24 operators; lower is better.

Method	Iterations
Evolution AMG, one cycle	20.08
Evolution AMG, two cycles	13.63
Nearest training target	18.13
Four-neighbor target interpolation	18.27
Solver-aligned FNO	18.97

Table 18: Outside the rotated-anisotropy family, two screens stop before training and no model trained after a passed screen passes its frozen validation gate. Iterations are equal-operator means; lower is better. "Target" is the exact finite solver target around the same base preconditioner.

Family	Target screen	Prediction
2D lognormal diffusion, contrast 10^4	Passed: Evolution AMG 21.91, target 13.45, two cycles 15.28 (8 operators)	Failed: on 12 validation operators the solver-aligned FNO gives 22.95 against 21.34 for one cycle (-7.91%) and 14.14 for its target; it is worse on all 12. Energy-norm and worst-direction losses give 20.25 and 20.02 against one cycle 19.80 in a shared-hierarchy replay.
3D obstacle family, large FNO with energy losses	Passed (Table 15)	Failed gate: the 37.8M-parameter FNO trained with the mean and worst-direction energy losses gives 18.15 and 18.10, against 18.81 for the Euclidean loss, 18.04 for the Rayleigh FNO, 24.75 for one GMG3D cycle, and 14.33 for the target.
2D random checkerboard orientation	Stopped: the family is not uniformly SPD (indefinite target Gram matrix at operator 5)	Not trained.
HYPRE elasticity fixture	Passed: an exact scaled rank-16 space lowers BoomerAMG from 23.13 to 9.0; stopped because a classical adaptive rank-16 space also reaches 9.0	Not trained: no role for learning.

Removing the source channel also leaves solver behavior unchanged. With one shared uncorrected Evolution AMG baseline, the original and operator-only checkpoints both reduce the mean count from 32.67 to 26.58 and produce identical corrected counts on all 12 operators. Two additional operator-only training seeds give the same 12 corrected counts.

The BoomerAMG audit uses four right-hand sides on each of eight development operators to choose among defaults and strength thresholds 0.5, 0.7, and 0.9. Threshold 0.5 has the lowest development mean, but on the 12 test operators its uncorrected and corrected counts are 10.60 and 10.34. The paired mean gain is 0.26 iterations with descriptive interval $[-0.02, 0.53]$; learning improves four operators and ties eight. The default comparison is stronger at 10.04 versus 9.63, but remains a small absolute effect. On all 12 test operators, an independent audit applies each one-cycle BoomerAMG map to four random vector pairs. The largest relative bilinear-form asymmetry is 1.01×10^{-12} for defaults and 7.56×10^{-13} for the selected setting; every sampled quadratic form is positive. The audit also records the resolved HYPRE cycle and relaxation options, supporting PCG compatibility rather than assuming it from defaults. Applying the same probes to the complete mixed-precision balancing map gives maximum asymmetry 9.92×10^{-5} for defaults and 4.82×10^{-5} for the selected setting; all sampled quadratic forms are positive. This is finite-precision support, not an exact symmetry proof. The separate precision audit gives identical float32 and float64 stopping counts on all 192 matched solves.

Table 19: Learned prediction beats fixed local and universal controls but not nearest-target retrieval on this narrow family. Paired rank-32 controls around Evolution AMG. Counts average 16 right-hand sides within each of 12 operators. Δ is method minus learned with a descriptive 95% interval; M/T/L counts method wins, ties, and learned wins. Dim./local gives total dimension and GenEO modes per subdomain. Exact Rayleigh space is nondeployable. A single dimension denotes a method without a domain partition.

Method	Dim./local	Iterations	Δ [95% interval]	M/T/L
Evolution AMG	0	32.57	$5.89\ [4.51, 7.26]$	$0/0/12$
Universal training-solution POD	32	30.86	$4.18\ [3.26, 5.10]$	$0/0/12$
GenEO-derived, 4×4	$32/2$	32.49	$5.81\ [4.47, 7.16]$	$0/0/12$
GenEO-derived, 2×2	$32/8$	31.14	$4.46\ [3.51, 5.41]$	$0/0/12$
GenEO-derived, 1×1	$32/32$	28.15	$1.47\ [-0.95, 3.88]$	$7/1/4$
Learned	32	26.68	0.00	$0/12/0$
Nearest training target	32	26.41	$-0.28\ [-0.59, 0.04]$	$9/1/2$
Exact Rayleigh space	32	26.39	$-0.29\ [-0.61, 0.03]$	$9/1/2$

Table 20: Adaptive GenEO-derived spaces can surpass the learned rank-32 space by using substantially larger dimensions. Separate single-right-hand-side sweep around Evolution AMG on the 12 rotated-anisotropy operators. Dimensions vary by operator and are reported as means.

Method	Mean dimension	Iterations
Evolution AMG	0	32.58
Learned	32	26.58
Adaptive GenEO-derived, $\tau = 20$	113.7	26.17
Adaptive GenEO-derived, $\tau = 10$	208.3	20.42
Adaptive GenEO-derived, $\tau = 5$	327.2	16.25

A later frozen development screen asks whether aligning the target itself to the selected BoomerAMG map leaves enough headroom to justify another training run. It uses eight new operators, eight matched right-hand sides per operator, and the same rank-32 correction. The 64-step finite target improves all eight operators and lowers the equal-operator mean from 11.047 to 9.500 iterations, a 14.00% reduction. Two residual-correction applications of the unchanged hierarchy average 7.469 iterations, however, so the target-to-two-cycle ratio is 1.272 and the prespecified criterion fails. Target construction averages 33.2 seconds per operator in this diagnostic run. This comparison is in outer iterations, not matched work: each two-cycle iteration applies the hierarchy twice. No network is trained, and reserved validation and test streams are not accessed.

Table 21 gives two complete repeated-right-hand-side timing frontiers. Each ratio first averages three alternating-order repeats within an operator, then takes a geometric mean across 12 operators. Intervals are descriptive Student-t intervals on operator log ratios. The machines use the same GPU model and software protocol, but timings are never pooled. These runs wrap evolution-strength Ruge–Stüben AMG; they are not timing comparisons with the lower-iteration BoomerAMG configuration in Table 1.

C.2 Numerical-choice and data controls

Table 22 expands the controls summarized in Section 3.3. They use the same held-out operator streams within each comparison, but answer different questions and are not pooled into one benchmark score.

Figure 7 places representative controlled and public inputs in one visual summary. Tensor glyphs show principal direction and relative anisotropy; spatial panels show central slices. The gallery communicates structural diversity rather than a common physical scale.

The Mendeley control symmetrizes the 214 reported tensors as recommended by the dataset documentation (Xiong et al., 2024), verifies positive definiteness, and evaluates an anisotropy-enriched

Table 21: Learned execution reaches break-even only at large repeated-solve budgets, and the crossing varies by machine. Cold-to-learned wall-clock ratio including setup and cumulative solves on two machines. Values above one favor learned execution.

	Machine 1		Machine 2	
Right-hand sides	Ratio	95% interval	Ratio	95% interval
1	0.739	[0.670, 0.814]	0.872	[0.824, 0.922]
2	0.811	[0.755, 0.871]	0.919	[0.883, 0.956]
4	0.883	[0.839, 0.929]	0.962	[0.934, 0.991]
8	0.932	[0.898, 0.967]	0.994	[0.971, 1.017]
16	0.973	[0.946, 1.002]	1.013	[0.993, 1.034]
32	0.997	[0.975, 1.021]	1.024	[1.005, 1.043]
64	1.010	[0.991, 1.030]	1.030	[1.012, 1.048]

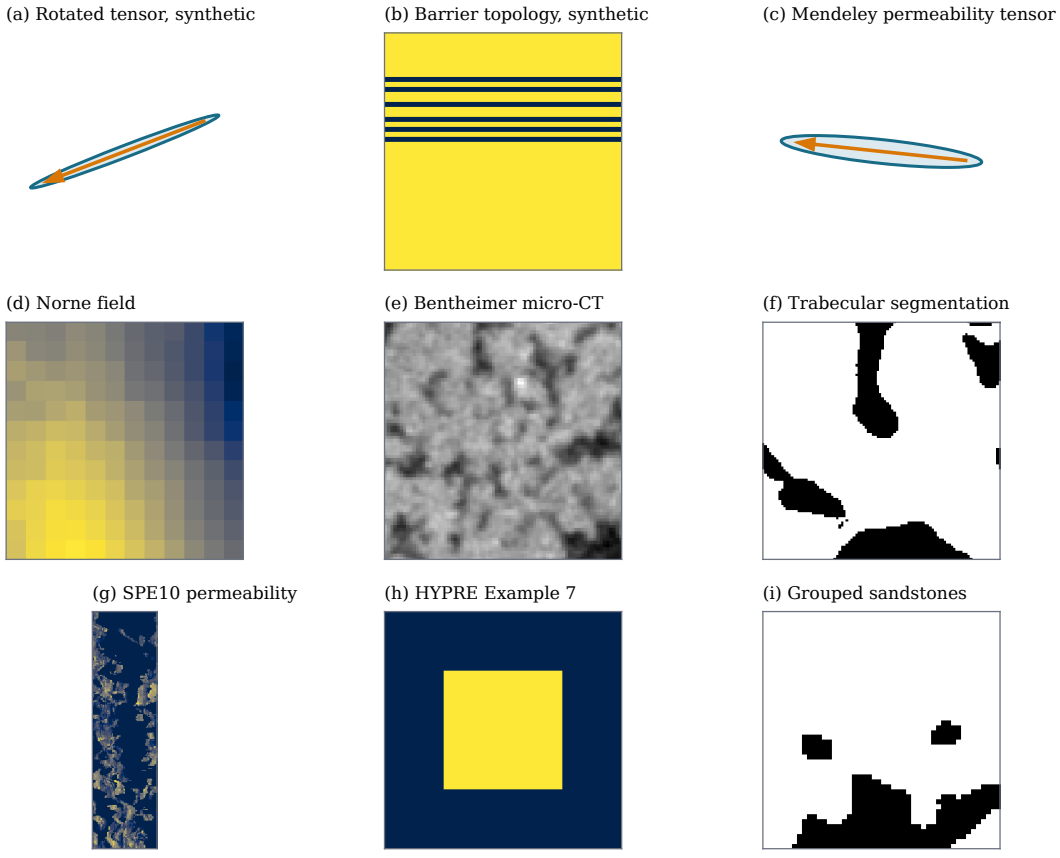

Figure 7: The evaluation spans controlled synthetic fields and structurally diverse public inputs. (a) Rotated-anisotropy tensor, (b) 3D barrier field, (c) public Mendeley tensor, (d) Norne permeability, (e) Bentheimer micro-CT, (f) trabecular segmentation, (g) SPE10 permeability, (h) HYPRE Example 7 coefficient, and (i) an 11 Sandstones crop. Glyphs encode principal tensor directions; spatial panels show representative central slices and use separate color scales.

40-projection subset. Uncorrected and corrected AMG average 10.88 and 8.58 iterations, but the median improvement is zero because 26 projections tie; the mean is therefore not a typical-case transfer effect.

Table 22: The 2D solver ordering is sensitive to numerical choices and does not carry unchanged to the tested 3D tensors. Counts are mean outer iterations except for the single Norne subvolume. The comparator columns identify the matched methods rather than a universal ranking.

Control	Method 1	Method 2	Comparison
Smoothed aggregation, 2D	82.58	68.83	SA / learned + SA
Geometric spaces, 2D	99.17	95.67	aggregation / bilinear
Line relaxation, 2D	46.67	167.67	standard / line GMG
Semicoarsening, 2D	46.67	63.33	standard / semicoarsened GMG
PyAMG finite difference, 2D	53.08	33.42	AMG / GMG
PyAMG finite element, 2D	39.75	43.83	AMG / GMG
Rotated tensor, 3D	20.40	28.00	Evolution AMG / GMG
Mendeley tensors, 3D	5.30	11.30	AMG / GMG
Norne geometry, 3D	5	16	AMG / GMG

C.3 Matched NeurKItt-style supervised objective

The objective control implements the exact-subspace supervision used by the closest FNO mechanism, NeurKItt (Luo et al., 2024), with the same operator-only inputs, FNO architecture, rank, 200 training operators, 80 epochs, training stochastic seed, normalized coordinates, deployment map, 12 test operators, and 16 right-hand sides per operator. The supervised variant replaces Eq. (3) with projection loss to the exact bottom-32 eigenspace. Within each test operator, the cold, unsupervised, and supervised solves share one seeded base hierarchy. Table 23 shows that supervision improves default AMG by 1.19 iterations on average. The advantage is smaller with stronger references and its descriptive interval includes zero with GMG. All 1,728 solves converge with recomputed relative residual below 10^{-8}. We do not report training wall-clock from these runs because they were not collected in a clean matched timing session. Thus the label-free objective's eigensolve saving is algorithmic but not quantified as a measured training-time advantage here.

Table 23: Exact-subspace supervision gives only a small advantage over the label-free objective on this family. Matched comparison on 12 rotated-anisotropy operators and 16 right-hand sides per operator. Cuts use the same uncorrected solver. The last column is the mean paired difference over the 12 operator-level means, unsupervised minus supervised, with a descriptive 95% Student-t interval. Positive values favor supervision.

Base preconditioner	Unsupervised cut	Supervised cut	Paired difference
Default AMG	47.9%	49.0%	1.19 $[0.35, 2.03]$
Evolution AMG	18.1%	19.4%	0.42 $[0.14, 0.71]$
GMG	7.5%	7.8%	0.16 $[-0.12, 0.44]$

D 3D topology details

D.1 Multi-seed synthetic result

Uncorrected AMG averages 8.2 iterations on the seed-7 suite. Uncorrected GMG3D averages 27.0, and the matched trained model averages 21.1. The learned correction therefore improves GMG3D without becoming the best solver. Training on 200 instances improves the matched corrected count to 20.5, a 24.1% cut.

D.2 Matched 3D retrieval control

The released retrieval protocol reconstructs training streams 1, 2, and 3 with 64 operators each and evaluation streams 7, 8, and 9 with six targets each. Barrier masks are compared by Jaccard distance, with exact rational arithmetic used to resolve distance ties. For every target and training stream, the selected donor contributes its exact normalized bottom-16 eigenspace, which is then mapped using

the target diagonal and reorthonormalized. This deliberately gives retrieval exact donor modes rather than an approximate cached prediction.

Uncorrected GMG3D, the corresponding FNO prediction, and all three retrieval spaces share each target operator, hierarchy, and 32 right-hand sides. The resulting 4,032 solves all satisfy the convergence and recomputed-residual gates. Target-level means are the independent units: cold, learned, and retrieval counts are 24.68, 18.58, and 23.12. Retrieval minus learned is 4.54 iterations with a 100,000-resample paired bootstrap interval of $[3.94, 5.10]$; learned is lower on all 18 targets. Pooling all 192 donors and choosing the globally nearest donor gives 23.28 iterations. Even a nondeployable oracle that retrospectively chooses the best of the three donor spaces gives 22.02; prediction remains lower on all 18 targets in both comparisons. Nearest Jaccard distances range from 0.424 to 0.802, with median 0.634. This supports prediction over nearest-training reuse within the synthetic 3D obstacle family only; it is not a public-transfer or wall-clock claim.

The development-tuned GMG3D control evaluates every combination of $n_{\min} \in \{4, 8, 16\}$ and $\nu \in \{1, 2, 4\}$. Increasing ν lowers development iterations but increases measured batch cost. At $n_{\min} = 8$, mean iterations are 34.72, 26.41, and 21.16, while median setup-plus-eight-solve times are 2.83, 3.14, and 3.84 seconds. The time-selected $\nu = 1$ configuration is evaluated with the same sequential CG recurrence on both sides of every held-out comparison; all 144 solves pass the true-residual gate.

Because time selection alone does not establish fairness for an iteration claim, a separately frozen sweep fixes $n_{\min} = 8$ and evaluates both uncorrected and corrected GMG3D at all three smoothing counts on the same nine held-out operators. Table 24 retains every result. Increasing smoothing is itself powerful: uncorrected $\nu = 4$ beats corrected $\nu = 1$. Within each matched setting, however, learning improves every operator and its percentage reduction is stable at 22.2% to 22.8%. Thus smoothing and correction are complementary; the experiment does not support comparing a weak uncorrected setting against a corrected strong setting or vice versa.

Table 24: Learning improves every matched smoothing configuration, while extra smoothing remains a strong classical intervention. Frozen held-out GMG3D sweep at $n = 48$ and $n_{\min} = 8$. Each mean first averages eight right-hand sides within each of nine operators. Differences are uncorrected minus corrected with descriptive 95% Student-t intervals over operators.

Sweeps	Uncorrected	Corrected	Difference [95% interval]	Learned wins
1	33.97	26.22	7.75 [5.94, 9.56]	9/9
2	25.64	19.94	5.69 [4.43, 6.96]	9/9
4	20.31	15.71	4.60 [3.74, 5.46]	9/9

The operator-dependent control uses the same nine operators, right-hand sides, checkpoint, and balancing correction. Both Ruge–Stüben and smoothed aggregation construct interpolation from the matrix rather than the fixed trilinear grid map. Table 25 reports every paired result; no timing fields were collected.

Table 25: The learned correction helps both tested operator-dependent AMG hierarchies on the synthetic family. Frozen 3D control; differences are uncorrected minus corrected with descriptive 95% Student-t intervals over nine operators.

Hierarchy	Uncorrected	Corrected	Difference [95% interval]	Learned wins
Ruge–Stüben	8.14	7.50	0.64 [0.36, 0.92]	9/9
Smoothed aggregation	21.11	16.81	4.31 [3.50, 5.12]	9/9

Early development used geometry seed 7 to choose architecture and training length, so results on geometry streams 7 through 9 are retrospective. The confirmatory protocol then moves to geometry seeds 101, 103, and 107, which had not been used for prior model or solver development. It evaluates all three checkpoints and a two-cycle classical control. A corrective frozen audit adds two boundary-compatible equal-rank controls: the 16 lowest tensor-product Dirichlet sine modes, directly and after inverse-diagonal mapping. A separate rank-16 classical frontier applies one, two, or four steps of subspace iteration on the AMG error propagator, costing 16, 32, or 64 setup V-cycles

per operator. Table 26 gives the iteration result. Every timed repetition rebuilds the hierarchy and correction; learned paths also rerun inference and QR. Method order rotates across three repeats. These warm-loaded times exclude checkpoint loading, normalizer reconstruction, operator assembly, and analytic-basis generation. The 32-right-hand-side cold-over-learned ranges are 0.922 to 0.930 at $n = 48$ and 0.967 to 0.979 at $n = 128$.

Table 26: Learning beats equal-rank controls but not an additional AMG cycle. Held-out operator-dependent 3D confirmation; learned is averaged over three checkpoints. Differences are control minus learned with descriptive 95% Student-t intervals over nine operators.

Method	Iterations	Difference [95% interval]	Learned wins
Uncorrected AMG	7.82	$0.87\ [0.70, 1.03]$	$9/9$
Diagonal sine rank 16	7.79	$0.84\ [0.63, 1.05]$	$9/9$
Dirichlet sine rank 16	7.57	$0.62\ [0.32, 0.91]$	$9/9$
Adaptive, 16 setup cycles	7.81	$0.85\ [0.67, 1.04]$	$9/9$
Adaptive, 32 setup cycles	7.81	$0.85\ [0.68, 1.03]$	$9/9$
Adaptive, 64 setup cycles	7.01	$0.06\ [-0.26, 0.38]$	$4/9$
Two AMG cycles	4.94	$-2.01\ [-2.29, -1.73]$	$0/9$
Learned rank 16	6.95	0.00	reference

A separate numerical-validity audit samples every one of the 27 learned operator-checkpoint balancing maps. The maximum relative bilinear asymmetry is 3.74×10^{-8}, and the minimum sampled quadratic form is 0.03395. A stronger intervention replays all 216 learned solves with float64 coarse arrays and factorization. No stopping count changes, and the largest true relative residual is 8.90×10^{-9}. Thus the reported counts do not depend on the released mixed-precision balancing arithmetic.

The selected configuration is then carried unchanged to three $n = 256$ maze operators and all four Bentheimer scans, with eight right-hand sides per operator and no timing claim. Table 27 reports the complete result. The small synthetic reduction is not resolved by the three-operator interval, whereas the correction is consistently harmful on the four public scans. The corresponding mean changes are a 5.64% reduction and a 5.92% worsening. Plain AMG remains the strongest method in both suites.

Table 27: The selected correction weakens at higher synthetic resolution and worsens all tested Bentheimer scans. Frozen transfer of the development-selected GMG3D configuration. Counts first average eight right-hand sides within each unit. The paired difference is selected uncorrected GMG3D minus corrected GMG3D, so positive values favor learning.

Suite	Units	AMG	Selected GMG3D	Corrected	Difference [95% interval]
Maze, $n = 256$	3	18.00	42.08	39.71	$2.38\ [-2.32, 7.07]$
Bentheimer	4	13.03	154.72	163.88	$-9.16\ [-13.93, -4.38]$

Table 28 isolates optimization variability by fixing the training data at seed 1 while changing initialization and epoch ordering. Each cell again averages six operators and 32 right-hand sides per operator. All 1,152 new learned solves satisfy the recomputed true-residual tolerance. The narrow spread of row means shows that the positive GMG3D correction is not specific to one training stochastic stream.

The predicted-space correction remains useful at these scales, but its earlier timing comparison with sequential AMG used asymmetric batching and is not a solver-speed claim. A "sieve" topology with many openings also has corresponding iteration headroom. These mechanism findings motivate better training but do not alter the transfer trend in Figure 4a.

A separate $n = 256$ uncorrected control uses the same CPU-batched GMG3D implementation, operators, right-hand sides, and stopping rule as the learned path. Its stopping counts equal the sequential GMG3D counts for all 24 right-hand sides. Timing fields are excluded because the runs were not repeated and interleaved under a continuously audited quiet host.

Table 28: The synthetic-family gain is stable across the three training stochastic streams. Matched GMG3D iteration cuts with the training-data stream fixed. Seed 0 is the released rank-16 Rayleigh FNO trained on data stream 1; seeds 1 and 2 change network initialization and epoch ordering.

Training seed	Eval. 7	Eval. 8	Eval. 9	Mean
0	21.94%	25.40%	28.13%	25.15%
1	20.74%	24.77%	28.92%	24.81%
2	20.91%	25.94%	27.31%	24.72%

D.3 Bentheimer diagnosis

To separate optimization failure from objective mismatch in 3D, we compute the exact Rayleigh space and finite solver targets at ranks 16, 32, and 64 on each Bentheimer operator around GMG3D. The normalized eigenspace is computed to residual below 10^{-7} and transformed exactly as at deployment; the finite solver target is evaluated after 15, 25, and 40 subspace iterations on the GMG3D error propagator. All comparisons keep per-right-hand-side stopping counts and use the same correction, operators, and batches. The exact Rayleigh space cuts GMG3D iterations by 24.5%, 71.7%, and 77.6% at ranks 16, 32, and 64; the finite solver targets cut them by −9.4%, 2.6%, and 7.2% at rank 16 (15, 25, 40 iterations), 45.8%, 65.6%, and 72.6% at rank 32, and 77.7%, 80.1%, and 80.3% at rank 64.

At rank 64, a Rayleigh FNO trained in-sample attains a lower Rayleigh trace than a solver-aligned FNO but reduces GMG3D iterations by only 16.9%, against 74.3% (Figure 8). The exact Rayleigh space and finite solver target both work well at this rank, while random and geometric spaces do not. Rayleigh value can therefore identify a useful exact space without reliably ordering approximate learned spaces. These are mixed in-sample and reference diagnostics, not a held-out comparison.

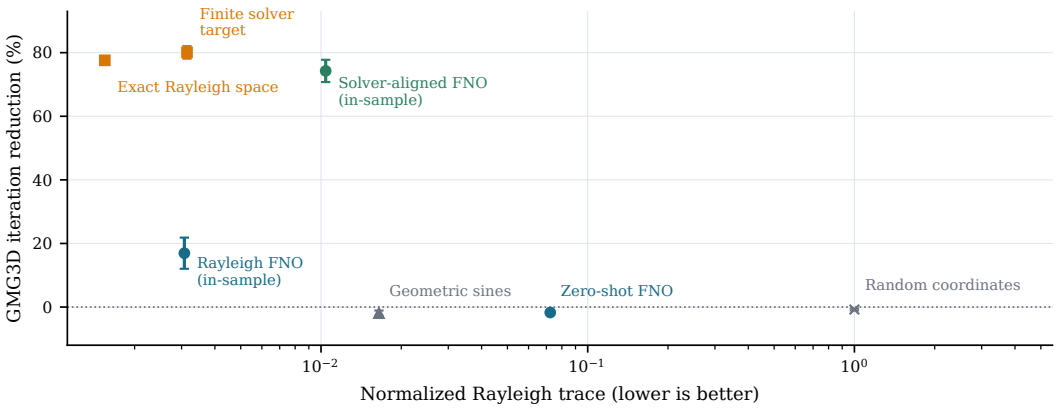

Figure 8: Rayleigh loss is not a solver metric. Points summarize three Bentheimer subvolumes at rank 64; bars are sample standard deviations. Marker shape distinguishes learned predictions from reference spaces. Lower trace does not consistently imply larger GMG3D iteration reduction.

On the complete four-scan subset, exact rank-64 Rayleigh averages 25.59 iterations while plain AMG averages 13.06 on the same operators and right-hand sides. The corresponding four-scan Rayleigh cut around GMG3D is 77.9%, so this remains a mechanism diagnostic rather than a solver win.

Exact Rayleigh setup takes tens of seconds per operator in this diagnostic implementation, as does error-propagator subspace iteration. These are diagnostic costs, not deployment claims. The 15, 25, and 40 iteration sensitivity is reported because the lower-rank finite solver targets are not numerically stable enough to justify "oracle" or "ceiling" terminology.

At rank 16 with 15 iterations, the finite solver target raises GMG3D iterations (a −9.4% mean per-operator cut in the rank sweep; the aggregate means in this run, 122.28 against 111.45, differ by 9.7%). In exact arithmetic balancing cannot cause this through a loss of contraction: if $H_1 \preceq I$, ev-

ery eigenvalue of the corrected operator lies in $[\lambda_{\min}(H_1), 1]$ (Eq. (4)). With single-precision coarse arrays the operator built from the stored factors is SPD, but rounding each residual to single precision makes the applied map slightly nonlinear, and the interval is no longer guaranteed. GMG3D, a symmetric weighted-Jacobi V-cycle with an exact coarsest solve, satisfies the argument of Appendix B.1 when each Jacobi sweep is an energy contraction. A post hoc diagnostic without training examines the anomaly on the three operators. With double-precision coarse arrays ($\mathbf{Z}$, $\mathbf{AZ}$, and the coarse Cholesky factor, stored in single precision at deployment) the finite solver target averages 115.28 iterations instead of 122.28, against 111.45 for GMG3D alone, so single precision accounts for most of the increase; we did not diagnose the cause of this sensitivity. In double precision, 120-step Lanczos gives an estimated $\lambda_{\max}(M_1\mathbf{A}) = 0.99998$ on every operator, and every rank-16 space lowers the estimated condition number, from 507–654 to 341–490 for the finite solver target. Iterations still rise on two of the three operators, and on one operator the finite solver target has a lower estimated condition number than the exact Rayleigh space (396 versus 426) but needs 122.0 rather than 91.4 iterations. The remaining increase is therefore consistent with CG behavior beyond the condition number, not with a loss of contraction; CG stops on the recursive unpreconditioned relative residual. The cold and exact Rayleigh means reproduce the rank sweep exactly; one of the six finite-target means differs from it by 0.69 iterations, above the prespecified 0.5 tolerance, so by protocol this run is not treated as a replication of the sweep and all comparisons in this paragraph are within the run.

Figure 8 reports the rank-64 aggregate comparison. The random-space deviation is over three operator means, each averaging three random draws.

The optimization path separates loss reduction from solver improvement. Mean loss is already 0.00509 at step 100 while the iteration change is negative; at step 500, loss is 0.00376 and the cut reaches 7.5%. The final direct space has mean overlap 0.643 with exact Rayleigh. Each operator has median principal cosine at least 0.963 but minimum cosine below 0.0014, indicating broad alignment alongside at least one poorly represented direction.

D.4 Preprocessing sensitivity

We compare resizing grayscale scans before Otsu segmentation with two variants that segment first and resize the mask; all retain a dominant pore component and axis-spanning connectivity. Solver difficulty changes across pipelines, but adding the learned correction increases the GMG3D count for all three, so the negative transfer conclusion is unchanged.

Table 29: Preprocessing changes solver difficulty but does not rescue transfer. Entries average three subvolumes and 32 right-hand sides per subvolume. The final column reports the mean after adding the learned correction to GMG3D.

Pipeline	Porosity	AMG	GMG3D	Corrected GMG3D
Resize gray, then Otsu	0.273	12.83	111.45	116.38
Otsu, then nearest mask	0.265	13.05	147.19	156.48
Otsu, then interpolated mask	0.260	13.21	120.83	125.74

D.5 Held-out public topology suite

The external protocol includes every filename ending exactly in `_SEG_SUB.nii` in Zenodo record 11061947 and excludes all derived distance transforms. Seven sources are 100^3 and one is 50^3; each has labels $\{0, 127\}$. The segmented-bone fraction after fixed nearest-neighbor resizing ranges from 0.098 to 0.648. Subject labels retain the source numbering; the public record has no matching subject 6 file, rather than an excluded sixth subject. Across subjects, the paired reductions are -2.73% for AMG and -4.79% for GMG3D.

The separate SPE10 protocol pins an immutable Open Porous Media revision before the permeability array is fetched. It retains the complete Model 2 field and the published Tarbert and Upper Ness formations without cropping or resizing. Because the released FNO accepts one scalar coefficient, the evaluator maps the three directional permeabilities to their geometric mean, retains the upper

Table 30: The fixed rank-16 Rayleigh FNO fails to improve either preconditioner consistently across public trabecular subjects. Each entry averages the same eight right-hand sides. No subject is excluded or used for model selection.

Subject	Bone fraction	AMG	Corrected AMG	GMG3D	Corrected GMG3D
1	0.296	8.00	8.00	41.12	43.75
2	0.284	8.50	8.75	33.62	33.50
3	0.371	10.00	10.75	51.12	55.88
4	0.193	7.12	7.12	17.75	17.88
5	0.098	7.25	7.00	16.12	16.62
7	0.390	10.00	10.00	52.25	56.25
8	0.648	12.75	13.25	108.00	121.50
9	0.513	8.00	8.88	26.38	26.12

four decades, and normalizes the maximum to one. This clips 50.3% of cells to 10^{-4}. Each row of Table 31 averages the same eight right-hand sides. Complete and formation rows are related views of one geological model rather than independent samples. Relative to uncorrected AMG, the mean changes are -0.53% on Tarbert and -0.07% on Upper Ness.

Table 31: Fixed rank-16 Rayleigh FNOs leave the SPE10 scalar proxy essentially unchanged. Frozen transfer at native array resolution; the three corrected columns are Rayleigh FNOs from independent 3D obstacle training streams.

Operator	Unknowns	Uncorrected AMG	Corrected 1	Corrected 2	Corrected 3
Complete	1,122,000	44.875	44.875	44.875	44.875
Tarbert	462,000	39.375	39.875	39.500	39.375
Upper Ness	660,000	59.625	59.625	59.750	59.625

D.6 SOURCE-VALIDATED HYPRE EXAMPLE 7

We prospectively froze an outcome audit of all four diffusion choices in the public HYPRE Example 7 source under $B = C = U_0 = 0$, using all four nonzero published right-hand sides. The one-process 258^2 SStruct system includes boundary nodes; eliminating its homogeneous boundary leaves the model's native 256^2 interior grid. A minimally instrumented copy of the pinned upstream source dumps the assembled matrix and right-hand side before solving. The source-faithful PETSc reconstruction agrees with those native objects to 5.33×10^{-14} in maximum matrix-entry error and 5.01×10^{-19} in maximum right-hand-side error.

Table 32 reports fixed-case descriptive means. Each entry averages four right-hand sides. The common-preconditioner rows use PCG tolerance 10^{-8}, one explicitly configured BoomerAMG V-cycle per outer iteration, and a recomputed true-residual ceiling of 10^{-7}. The learned and sine corrections have rank 32 and use float64 balancing arithmetic. Two residual-correction cycles reduce outer iterations further, but average 12.250 BoomerAMG applications rather than 6.125, so that row is not a cost comparison. The native SMG and PFMG rows retain the upstream 10^{-6} stopping tolerance and provide native-solver context only. Deterministic linearity, bilinear-symmetry, positive-form, and projected-spectrum probes provide empirical PCG-compatibility evidence rather than a full-space proof.

The learned correction gives a 4.11% equal-case reduction over one-cycle BoomerAMG, but the preregistered success criterion is not met. It ties K_0 and K_3, worsens K_2 by 0.250 iterations, improves only K_1, and is weaker than the scaled sine control. For K_0, F_2, the right-hand side is exactly aligned with the sine space's first mode, explaining that control's one-iteration solve; the sine control also remains competitive on the other cases. All common-preconditioner solves converge and pass the true-residual gate. This source-validated public benchmark with native controls reinforces, rather than resolves, the paper's transfer limitation.

Table 32: The learned correction does not beat simple sine spaces or a second BoomerAMG cycle on HYPRE Example 7. Columns K_0 through K_3 are the four published diffusion choices. The final column weights these fixed cases equally.

Method	K_0	K_1	K_2	K_3	Mean
BoomerAMG, one cycle	6.000	10.750	9.750	10.000	9.125
Learned rank 32	6.000	9.000	10.000	10.000	8.750
Sine rank 32, scaled	4.000	7.500	9.500	9.000	7.500
Sine rank 32, direct	4.000	7.750	10.000	9.000	7.688
BoomerAMG, two cycles	4.000	7.000	6.500	7.000	6.125
Native controls at the upstream 10^{-6} tolerance					
Native PCG with SMG	5.000	5.000	6.000	6.000	5.500
Native PCG with PFMG	9.000	9.000	10.500	9.000	9.375

D.7 Grouped 11 Sandstones study

We separately freeze a grouped study on all 11 segmented 1000^3 volumes in the public 11 Sandstones collection (Lucas-Oliveira et al., 2022). The seven groups combine named variants of Bandera and Berea and retain the five remaining rocks separately. Seven cyclic folds train on five groups, select one of four AMG configurations on a sixth group, and evaluate only the seventh. Each volume contributes 32 fixed nonoverlapping 48^3 crops. Subjects are averaged within group and the seven held-out groups receive equal weight; crops, 16 right-hand sides, and the single training seed are not independent units.

The frozen one-seed screen fails (Table 33). Selected one-cycle AMG averages 7.549 iterations and the Rayleigh FNO correction averages 7.994, a 5.83% worsening. Both right-hand-side families worsen: 6.65% for localized positive sources and 4.99% for smooth signed sources. The direct and inverse-diagonal-mapped rank-16 sine controls change the selected AMG count by -2.03% and -5.31%, respectively, while two cycles of the same selected hierarchy average 4.930 iterations. All 28,160 solves converge and pass the true-residual threshold. An exact rerun with the frozen producer implementation reproduces every iteration count and stored residual. The exact seven-group sign-flip tail probability is 0.0156, reported descriptively rather than as a population generalization claim. The 5.83% estimand is the arithmetic mean of the seven group-level percentage changes, not the percentage formed from the two displayed equal-group iteration means; the latter is 5.90%.

Table 33: The learned correction worsens every held-out sandstone group. Change is the paired iteration reduction relative to validation-selected one-cycle AMG, so negative values are worse. Groups, rather than subjects or crops, receive equal weight.

Held-out group	Selected AMG	Learned	Change
Bandera family	7.880	8.359	-6.09%
Bentheimer	7.514	7.938	-5.64%
Berea family	7.459	7.853	-5.28%
Castle Gate	6.771	7.111	-5.02%
Kirby	6.873	7.283	-5.97%
Leopard	7.205	7.596	-5.42%
Parker	9.141	9.818	-7.41%
Equal-group mean	7.549	7.994	-5.83%

Only the all-solves criterion passes. Zero of seven groups improve, the mean is below the required 5% improvement, every group exceeds the allowed 2% worsening, both source-family effects are negative, and the learned space does not beat both sine controls. Following the frozen policy, we stop without tuning on these groups, additional seeds, adaptive controls, retrieval, or timing. The result strengthens the distribution-shift diagnosis; it does not support external transfer.

E Related-work verification and public data

Table 34 separates the present contribution from NeurKItt (Luo et al., 2024), DeepONet preconditioning (Kopaničáková & Karniadakis, 2025; Kopaničáková et al., 2025), NSPOD (Levrero-Florencio et al., 2026), NeuraLSP (Benanti et al., 2026), graph coarse spaces (Liu et al., 2026), and RAPNet (Fink et al., 2026).

Table 34: The closest methods differ in predicted object, supervision, or base preconditioner. "Exact spectral" denotes eigensolve- or decomposition-derived supervision. Our original objective is label-free; the extension distills approximate slow spaces generated by the base preconditioner.

Work	Predicted object and signal	Distinction from this study
NeurKItt	Orthonormal invariant subspace, used to guide Krylov iteration, from exact eigenvector targets	Closest FNO mechanism; ours compares a label-free Rayleigh objective with approximate preconditioner-generated targets and audits the deployed multigrid map
DeepONet	Solution-informed subspace or correction from solution or reduced-basis data	Ours predicts a coarse basis from operator channels and leaves the first-level multigrid map unchanged
NSPOD	POD subspace via DeepONet from snapshot-derived POD targets	Includes BoomerAMG and PETSc smoothed-aggregation comparisons; ours centers matched baseline, retrieval, and transfer stress tests
NeuraLSP	Fixed-rank coarse basis from relaxed random test vectors and a self-supervised subspace loss	Uses test-vector inputs and a Galerkin two-level correction; ours predicts from operator channels and audits a fixed AMG or GMG preconditioner
Graph coarse	Meshfree multiscale basis via GNN from exact local spectral spaces	Targets two-level Schwarz and compares with its exact spectral construction; ours wraps unchanged AMG or GMG
RAPNet	Sparse AMG correction via GNN with a solver-derived objective	Modifies sparse AMG operators; ours predicts a dense low-rank space and studies when that addition helps
This work	FNO coarse basis from either Rayleigh trace or approximate ideal-solver-space overlap	Label-free original objective and solver-aligned extension; controlled baseline, budget, retrieval, resolution, and transfer audit

RAPNet (Fink et al., 2026) learns sparse AMG coarse operators with a GNN and includes a spatially varying 3D problem, but not our Ruge–Stüben, evolution-strength, GMG, and GenEO frontier. Klawonn et al. (Klawonn et al., 2026) learn adaptive domain-decomposition spaces from eigensolve-derived targets. Melchers et al. (Melchers et al., 2026) study replacing a coarse solve. We instead predict a basis for an unchanged balancing correction and stress-test its baseline, budget, objective, and transfer dependence.

The Mendeley tensors, Bentheimer scans, and trabecular-bone segmentations are CC BY 4.0 (Xiong et al., 2024; Jackson et al., 2021; Kuczynski, 2024); Norne and the Open Porous Media distribution of SPE10 use the Open Database License (Open Porous Media Initiative, 2026). The SPE10 model is documented by Christie & Blunt (2001). Each of the three deposits has an associated publication that we cite alongside the data DOI: the Bentheimer scans are released with Jackson et al. (2022), the trabecular-bone segmentations were produced to validate Kuczynski et al. (2023), and the Mendeley permeability and formation-factor tensors were computed with the FFT solvers of Cai et al. (2021), whose author list contains all four depositors. The artifact records each source and license. No proprietary material or internal terminology is used.