ENERGY-DECREASE TRAINING FOR LEARNED CONJUGATE DIRECTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learned iterative solvers are often trained without solution labels by minimizing the residual left after a correction. For a symmetric positive-definite system this under-rewards removing low-eigenvalue error: squared residual weights each error eigenmode by λ^2 while energy weights it by λ, so such error can be large in the solution yet small in the residual. We show that, in a learned conjugate-direction solver, changing only the training objective substantially changes how much of this error is removed. Energy-DCDM (E-DCDM) maximizes the exact energy decrease of each step, divided by the error energy visible on a small coarse space (32 sine modes); it needs no solution labels and leaves the deployed solver unchanged. When both losses train on the same solver states, E-DCDM lowers relative solution error after eight operator products by at least $3.3\times$ on every paired seed on public Darcy and screened-Poisson problems, both under a shared optimizer recipe (6.6 to $9.8\times$) and when each loss's learning rate and clipping are tuned separately (3.3 to $5.4\times$ on screened Poisson). Numerically verified Darcy eigenvectors indicate that energy training removes more low-eigenvalue error. The gain needs a suitable state weighting: in 2D, energy decrease weighted by $1/\|r\|_2^2$ is worse than the residual loss, and a generic coarse-free normalizer does not reproduce the 2D gain. E-DCDM can have higher residual error, the method and main comparison were designed after earlier methods were scored on the same evaluation data, and on 47 held-out 3D operators only a coarse-free variant was tested ($1.8\times$ lower one-step solution error). The study isolates the training objective rather than proposing a competitive solver: with stencil access, algebraic multigrid reaches far higher accuracy on Darcy within seven V-cycles.

1 INTRODUCTION

Neural operators amortize families of PDE solves into fast predictions (Li et al., 2020; Kovachki et al., 2023). When a prediction must satisfy a tight tolerance, it can initialize an iterative corrector rather than replace one. This pattern appears in residual correction (Cao et al., 2023; Jha, 2024), neural preconditioning (Rudikov et al., 2024; Li et al., 2025), and learned iterative solvers. The Deep Conjugate Direction Method (DCDM), for example, learns full-field search directions without solution labels by minimizing the residual left after an exact line step (Kaneda et al., 2022).

That observable objective is not neutral. Let the symmetric positive-definite matrix A define $Au = b$, let x be an approximate solution, and let $e = u - x$ be its error. For orthonormal eigenpairs $A\phi_i = \lambda_i \phi_i$, coefficients $c_i = \phi_i^\top e$, and energy norm $\|e\|_A^2 = e^\top A e$,

$$\|e\|_2^2 = \sum_i c_i^2, \qquad \|e\|_A^2 = \sum_i \lambda_i c_i^2, \qquad \|Ae\|_2^2 = \sum_i \lambda_i^2 c_i^2. \tag{1}$$

Energy gives the intermediate weighting. Low-eigenvalue error can dominate L^2 while contributing little to residual norm, a classical motivation for multigrid, deflation, and energy minimization. Our question is deliberately narrower: *if a neural network supplies conjugate search directions, how much does changing only its training objective matter?* In a controlled study of the objective, not of solver competitiveness, we find that it matters substantially.

We call our method Energy-DCDM (E-DCDM). For residual $r = b - Ax$ and a candidate deployed direction p, it maximizes the exact energy decrease $(p^\top r)^2/(p^\top Ap)$. Both losses use only residuals

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.

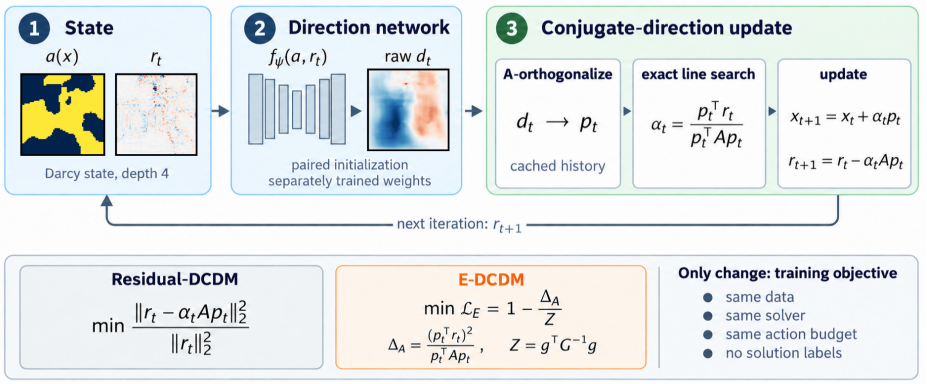

Figure 1: E-DCDM changes only the training loss inside a shared learned conjugate-direction solver. The network maps the current operator and residual to a raw direction; cached A-orthogonalization, exact line search, and the recurrent update are identical in both arms. The heatmaps show one real Darcy validation state at recurrent depth four; raw d_t is produced by the selected E-DCDM checkpoint, with residual and direction colors independently zero-centered and scaled.

and operator products; the matched Residual-DCDM baseline minimizes the post-line-step residual norm (Figure 1). Everything else is shared: the CNN, problem instances, per-seed initialization, optimizer (except in one separately tuned control), cached-span projection, exact line search, validation rule, and an eight-operator-product budget. Both losses also train on the same solver states: a fixed mixture of warm starts and later recurrent states with their cached histories (the *matched recurrent-state condition*), rather than warm starts alone (the *initial-state condition*).

Under these controls, changing only the loss lowers field error at least $3.3\times$ on every paired seed in both 2D families: 8.3–$9.8\times$ on Darcy and 6.6–$8.9\times$ on screened Poisson under the shared recipe, and 3.3–$5.4\times$ on screened Poisson after tuning each loss's learning rate and clipping (Section 5.1). The gain needs energy decrease, which the verified Darcy eigenmodes indicate removes more low-eigenvalue error, together with a suitable state weighting: energy decrease weighted by $1/\|r\|_2^2$ is worse than the residual loss, and the coarse normalizer did best (Section 5.2). On held-out 3D operators, only the coarse-free normalizer Z_{SD} was tested; it lowers one-step field error $1.8\times$. E-DCDM can have higher residual error: up to 25% early in 2D rollouts, $1.38\times$ after one 3D step, and under zero-shot grid refinement; both learned solvers fail at extreme coefficient contrast. The main method and comparison are post hoc (Section 6), and with stencil access, algebraic multigrid remains more accurate on Darcy (Section 5.4).

Our contributions are:

- **A drop-in loss.** A solution-free energy-decrease loss for learned conjugate directions, normalized by the error energy visible on a small coarse space (whose choice matters; Section 5.2); it changes nothing at deployment.
- **A controlled comparison.** Residual and energy training on identical recurrent states in two public 2D families, with paired seeds and an FNO architecture control, and, with a generic normalizer, 47 held-out 3D operators.
- **A mechanism and its limits.** An eigenmode audit and an objective-by-weighting factorial showing that the gain requires energy decrease and a suitable state weighting (the coarse weighting did best), plus stress tests where it fails.

2 Setting and Residual-Blind Error

For a coefficient field a, consider

$$A(a)u = b, \quad A(a) = A(a)^\top \succ 0, \quad e_t = u - x_t, \quad r_t = b - Ax_t = Ae_t. \tag{2}$$

The deployment interface exposes (a, r_t) and matrix-free products Av. Our primary metric is relative solution error, called *field error* below, $E_2 = \|u - x_t\|_2/\|u\|_2$. We also report relative energy error

$E_A = \sqrt{e_t^\top A e_t / (u^\top A u)}$ and relative residual $E_R = \|r_t\|_2 / \|b\|_2$. We count one *operator product* whenever the solver requests Av for a new v, including the one that forms r_0; network inference is timed separately. Each learned correction requests one new product Ap_t, so an eight-product learned rollout is the initial residual plus seven learned corrections; counts reported after the initial residual exclude the common first product. Exact targets are used only to evaluate E_2 and E_A, never by either training objective. Appendix A.1 details the online and offline product accounting and the operator-access setting.

Learned conjugate directions. Our comparator adapts the released DCDM architecture and objective to the 2D operators in this study. Given (a, r_t), a CNN predicts a full-field direction d_t. Full-memory deployment removes its A-projection onto every cached direction, giving p_t, requests Ap_t, and takes the exact line minimizer

$$\alpha_t = \frac{p_t^\top r_t}{p_t^\top A p_t}, \qquad x_{t+1} = x_t + \alpha_t p_t, \qquad r_{t+1} = r_t - \alpha_t A p_t. \tag{3}$$

The released-style recurrence retains two recent pairs; our stronger comparator retains every (p_j, Ap_j). Cached products make orthogonalization free in operator products, at $O(tn)$ vector storage and work after t steps for n unknowns. We retrain the released architecture for each operator and call it Residual-DCDM rather than an unchanged released checkpoint.

3 Energy training for learned conjugate directions

Let $d_\psi = f_\psi(a, r)$ be the raw output of the convolutional direction network used by DCDM, and let p_ψ be its A-orthogonal projection against the cached directions (Appendix A.1, Eq. 8). Both objectives are evaluated on this same deployed direction p_ψ, and gradients pass through the projection.

3.1 Exact energy decrease

The Residual-DCDM baseline minimizes the post-line-step residual ratio

$$\mathcal{L}_{\mathrm{R}}(p; r) = \frac{\|r - \alpha(p) A p\|_2^2}{\|r\|_2^2}, \qquad \alpha(p) = \frac{p^\top r}{p^\top A p}. \tag{4}$$

This is the released DCDM loss, adapted to our operators.

For an SPD operator, define the energy decrease obtained by an exact line search along p as

$$\Delta_A(p; r) = \frac{(p^\top r)^2}{p^\top A p}. \tag{5}$$

Equivalently, it is the decrease in squared A^{-1} residual norm: $\|r\|_{A^{-1}}^2 - \|r - \alpha A p\|_{A^{-1}}^2 = \Delta_A(p; r)$. By Cauchy–Schwarz in the A-inner product, $\Delta_A(p; r) \le r^\top A^{-1} r = \|e\|_A^2$, with equality precisely when $p \propto A^{-1} r = e$. Thus the solution-free objective targets the exact error direction up to scale, while weighting spectral error by λ_i rather than the residual objective's λ_i^2.

The exact relative post-step energy objective is

$$\frac{\|r - \alpha(p) A p\|_{A^{-1}}^2}{\|r\|_{A^{-1}}^2} = 1 - \frac{\Delta_A(p; r)}{r^\top A^{-1} r}. \tag{6}$$

Its denominator $r^\top A^{-1} r = e^\top A e$ is unavailable without a solve.

What differs between the losses. The objectives have the same unconstrained global solution: if the network could emit a projected direction $p = e = A^{-1} r$ for every training state, the exact line step would solve the system and make either loss optimal. The distinction appears under shared finite capacity and finite optimization. The residual loss is also a hybrid: it takes the energy-optimal (CG) step length and then measures the residual, unlike conjugate-residual or MINRES methods, which use the minimal-residual step (Saad, 2003). After an energy-optimal step the residual can grow while the error shrinks: for $A = \mathrm{diag}(1, 100)$, $r = (1, 1)$, and $p = (1, 0.1)$, squared energy

error drops 60% and squared L^2 error about 80%, while the squared residual norm rises from 2 to 20.45. The energy decrease Δ_A is exactly what the shared exact line step maximizes along p (Hestenes & Stiefel, 1952), so the residual loss scores a step by a criterion the step does not optimize. Concretely, because $r - \alpha_{\mathrm{MR}} Ap \perp Ap$ with $\alpha_{\mathrm{MR}} = r^\top Ap / \|Ap\|_2^2$, the residual loss splits into the relative squared residual left by the minimal-residual step along p and a step-overshoot penalty $(\alpha(p) - \alpha_{\mathrm{MR}})^2 \|Ap\|_2^2 / \|r\|_2^2$; Section 5.2 tests how much of the energy loss's advantage this penalty accounts for.

Remark 1 (single-mode priority). For $e = \sum_i c_i \phi_i$, an exact line step along ϕ_i has length c_i (since $\phi_i^\top r = \lambda_i c_i$ and $\phi_i^\top A \phi_i = \lambda_i$) and decreases squared field, energy, and residual errors by c_i^2, $\lambda_i c_i^2$, and $\lambda_i^2 c_i^2$. Residual geometry therefore assigns an extra factor λ_i when choosing among imperfect mode corrections; this is a priority statement for restricted candidates, not a dominance result.

3.2 Coarse normalizer

Because the denominator of Equation 6 requires a solve, we instead choose a fixed coarse basis V of 32 sine modes and form $G = V^\top A V$, $g = V^\top r$, and $Z = Z_{32} = g^\top G^{-1} g$. This is the energy captured by the Galerkin correction in $\mathrm{span}(V)$, that is, the error energy visible on the coarse space; for V of full column rank and $g \neq 0$, $0 < Z \leq r^\top A^{-1} r$, a computable lower bound on the total error energy. Equivalently, Z is the largest energy decrease of a single step within $\mathrm{span}(V)$, so the loss measures how much the learned step improves on the best coarse step. E-DCDM minimizes

$$\mathcal{L}_{\mathrm{E}}(\psi) = 1 - \frac{\Delta_A(p_\psi; r)}{Z}. \tag{7}$$

The denominator is constant with respect to the predicted direction and induces inverse-Z state weighting; it does not restrict the dense output to V. The numerator can exceed Z, so the loss is bounded below by $1 - r^\top A^{-1} r / Z$, which tends to $-\infty$ as the captured fraction $Z/(r^\top A^{-1} r)$ tends to zero. Appendix A.1 proves the bound and specifies the numerical safeguards. Both objectives request one Ap per training state (plus, for Z, a Gram matrix G formed once per operator offline), and neither reads a solution or error label. Because Δ_A is scale invariant, the line search sets the deployed magnitude.

We write E-DCDM (Z) when another normalizer Z replaces Z_{32}. The generic alternative $Z_{\mathrm{SD}} = (r^\top r)^2 / (r^\top A r)$ is the exact energy decrease of one steepest-descent step, the best step within $\mathrm{span}\{r\}$; it needs no coarse space but carries no coarse spectral information. Appendix Table 3 compares the normalizers used in this paper and their costs.

Two coupled ingredients. Within a training state, E-DCDM rewards the exact energy decrease Δ_A instead of the residual decrease (*objective geometry*). Across training states, the normalizer $1/Z$ sets how strongly each state competes for the shared network (*state weighting*).

3.3 Training on solver states

The loss applies to any differentiable direction network f_ψ; both primary arms use the same DCDM-style CNN (Appendix A.1), and we test an FNO separately. Our primary models train on the same fixed mixture of recurrent states from residual- and energy-trained trajectories at depths 0–6. This is state matching, not unrolled optimization: each stored state contains its cached history, and gradients traverse the projection of only the new raw direction. At every step the new raw direction is A-orthogonalized against all cached directions using cached products, so the projection needs no new call to A and is identical for the two losses. Since exact line steps keep the residual orthogonal to the cached directions, the projection preserves $p^\top r$ and does not increase $p^\top Ap$, so it cannot reduce the obtainable energy decrease Δ_A (Appendix A.1).

4 Relation to prior work

Closest objective comparisons. DCDM predicts solution-free full-field directions and takes exact line steps (Kaneda et al., 2022). We adapt its direction architecture and exact-step construction,

then give both paired arms the same strengthened full-memory recurrence; only their training loss differs. FCG-NO trains a nonlinear flexible-CG preconditioner on Krylov residuals and finds that a solution-supervised relative A-error loss beats supervised L^2 (Rudikov et al., 2024). CGiNS likewise finds that label-free minimum-potential-energy training of a 3D preconditioned conjugate-gradient (PCG) initializer beats a PDE-residual loss (Xing et al., 2025). Prior work therefore motivates energy-aware training, but does not isolate a solution-free normalized exact-step objective inside a matched recurrent-state learned-direction solver or connect the resulting corrections to verified numerical operator eigenmodes; this paper does both.

Learned iterative solvers. HyDEA alternates approximate-inverse DeepONet directions with CG (Bai & Bian, 2026); UNO-CG, NPO, learned two-level methods, and neural incomplete factorizations learn reusable preconditioners (Herb & Fritzen, 2026; Li et al., 2025; Melchers et al., 2026; Häusner et al., 2024). Other work differentiates through flexible GMRES (FGMRES), inserts an FNO into nonlinear CG, or trains residual-conditioned policies with reconstruction labels (Dimola et al., 2025; Wu et al., 2025; Jiang et al., 2026); see also Li et al. (2023) and Hsieh et al. (2019). With full memory, our learned recurrence is a flexible conjugate-gradient method whose preconditioner is a nonlinear network (Notay, 2000). In contrast to the works above, both of our arms use no solution labels, make the same nonlinear direction call at every step, preserve full-memory conjugacy, and differ only in one-step residual versus energy-decrease training.

Global-mode correction. Classical multigrid and deflation already target slow global modes (Nicolaides, 1987; Saad et al., 2000; Tang et al., 2009). Neural variants learn prolongation, low-frequency corrections, invariant subspaces, or reduced bases (Greenfeld et al., 2019; Luz et al., 2020; Kopaničáková & Karniadakis, 2025; Luo et al., 2024; Zheng et al., 2025). Residual correctors and warm-start hybrids instead learn around a neural solution map (Cao et al., 2023; Jha, 2024; Zhang et al., 2022; Eshaghi et al., 2025). Deflated or two-level CG with the same 32 modes and AMG are the natural classical alternatives (Section 5.4; Appendix A.8 reports a known-stencil 32-mode correction).

5 Experiments

Darcy. We form an SPD discretization of $-\nabla \cdot (a(x)\nabla u) = 1$ on public piecewise-constant coefficient masks (Li et al., 2020). Development fields come from `darcy_train_64.pt` (NeuralOperator 64×64 release); the 1,000 evaluation fields are the first 1,000 of `piececonst_r421_N1024_smooth2.mat`, subsampled to 64×64. No evaluation field matches a development field (sources in Appendix A.2). Evaluation crosses every coefficient with 16 independently trained FNO/U-Net warm-start models, grouped into four equally weighted warm-start cells (combinations of architecture and data root) (Takamoto et al., 2022).

Screened Poisson. $[-\Delta_h + 25 + 100m(x)]u = f$ on public coefficient/forcing instances from the convolutional neural operator (CNO) release (Raonic et al., 2023). Disjoint development and test populations form four balanced strata crossing in/out forcing with FNO/U-Net warm starts.

3D extensions. We test scale and operator variation. The first uses the DCDM authors' public 64^3 pressure systems, whose held-out right-hand sides share two simulation operators (Kaneda et al., 2022); the second constructs distinct 32^3 conductivity operators from public ModelNet10 meshes, with categories disjoint across splits (Wu et al., 2015). The latter evaluates five paired seeds on 47 held-out operators.

Comparators and evaluation. Paired networks share initialization, training states and order, optimizer, recurrence, and evaluator; each loss selects its own checkpoint by held-out objective without test targets. Primary 2D paired intervals resample shared test instances and condition on the trained networks; the estimand is the effect of replacing Equation 4 by Equation 7 under this recipe (Appendices A.1 and A.2).

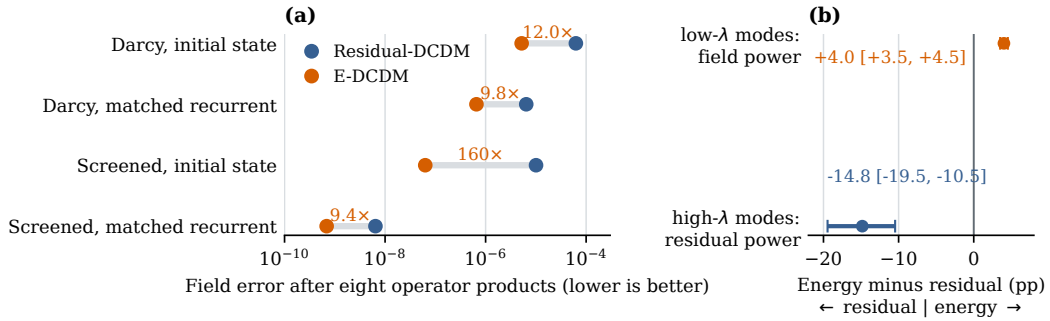

Figure 2: Matched recurrent states narrow but preserve the loss gap, and verified numerical Darcy eigenmodes show the predicted trade. **(a)** Complete losses within each state condition at the eight-product budget (validation-selected checkpoints; same-seed ratios in Section 5.1). **(b)** Difference in selected-mode power removed, energy minus residual geometry, in percentage points (pp); bars are 95% intervals clustered by 40 validation instances (construction in Section 5.2 and Appendix A.5).

5.1 The loss effect under matched recurrent states

Figure 2a is the central controlled result. Training only on warm starts may favor one loss, because deployment visits later states; the matched condition gives both losses the same states and cached histories (and twice the training states and updates). Both improve, yet E-DCDM remains $9.8\times$ lower in Darcy field error and $9.4\times$ lower on screened Poisson (validation-selected models), with the same sign for every instance, warm-start model, and balanced stratum; exact endpoints and paired intervals are in Appendix A.2.

With the official NeuralOperator FNO as the direction network, all instances and warm-start cells still favor E-DCDM (Darcy field error 1.28×10^{-3} for E-DCDM versus 9.38×10^{-2} for Residual-DCDM; both far above the CNN errors; Appendix A.4). For the CNN, every same-seed pair favors E-DCDM: Darcy 8.3–$9.8\times$ (geometric mean $9.2\times$) and screened Poisson 6.6–$8.9\times$ ($7.3\times$), so checkpoint selection does not explain the result. The instance-level intervals condition on the trained networks and do not cover this seed variation.

These comparisons share Adam at 10^{-4} with clipping. In a post hoc screened-Poisson control, each loss selected its learning rate (10^{-4} to 10^{-2}) and clipping from nine configurations (grid extended once; Appendix A.8) by mean validation objective; both selected 10^{-3} without clipping, an interior point. The same-seed gain falls to 3.3–$5.4\times$ (tuning lowers Residual-DCDM error $1.8\times$ and E-DCDM error $1.04\times$), and all nine fixed configurations favor E-DCDM on every seed. Darcy was not tuned; the transferred selections give 7.8–$9.0\times$.

Rollouts. All three errors decline at every product; E-DCDM has lower field and energy error throughout, but up to 25% higher residual error through product three (Darcy) or four (screened Poisson), and lower afterwards (Figure 6).

5.2 Why the loss matters: spectral priorities and the normalizer

Verified numerical eigenmodes support the mode priority of Remark 1. We compute verified extremal eigenvectors of the actual variable-coefficient Darcy operators on validation states. Each of three comparisons pairs the same initialization seed and the same additional inverse-Z weighting, isolating residual versus energy geometry up to the residual loss's own per-state $\|r\|_2^2$ denominator. Figure 2b uses *field power* for the sum of c_i^2 over the selected modes and *residual power* for the corresponding sum of $\lambda_i^2 c_i^2$. All 28 strata (architecture, source, and depth) favor energy geometry on low-mode field removal after averaging seeds, while 25 of 28 favor residual geometry on high-mode residual removal; every paired seed agrees in both directions. Appendix A.5 gives the numerical verification and stratum construction, and Appendix A.8 the fixed-sine control.

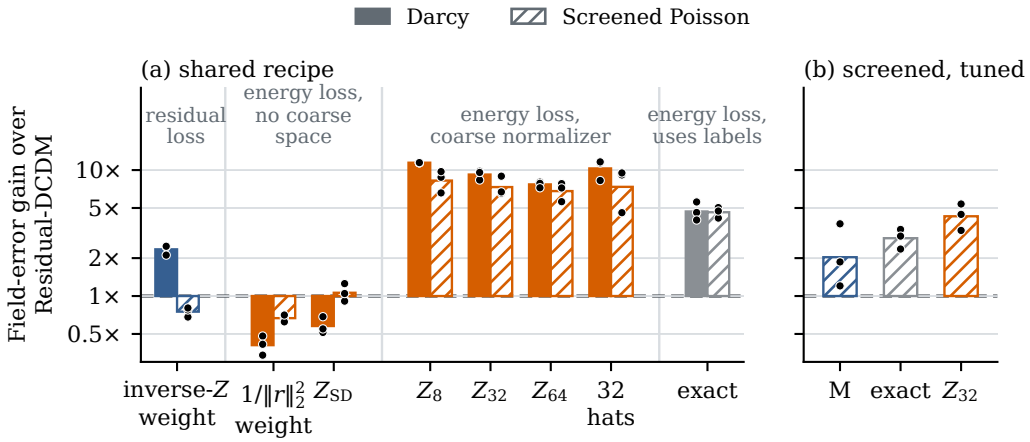

Figure 3: Among solution-free losses, only the energy loss with a coarse normalizer beats Residual-DCDM on both PDEs, and under matched states it also beats the solution-supervised exact normalizer $r^\top A^{-1} r$. Bars: field-error gain over Residual-DCDM after eight products (its error divided by the model's; geometric mean over three seeds; higher is better); dots: individual seeds, each against its same-seed Residual-DCDM; dashed line: equal error. Blue: residual loss; orange: energy loss; grey: uses solution labels. **(a)** Shared recipe; Z_{32} is E-DCDM; raw scaling (mean-matched in Appendix A.6). **(b)** Screened Poisson, each loss at its own tuned configuration, against tuned Residual-DCDM; M is the step-overshoot control (the residual loss without its overshoot penalty).

Objective geometry and state weighting interact. The denominator in Equation 7 does more than set a scale: it determines how recurrent states compete for the shared network. We therefore cross residual versus energy geometry with uniform versus inverse-Z state weighting on the same matched recurrent states (Appendix A.5, Equation 9); "uniform" keeps only each geometry's own per-state normalization, which for the energy arm weights states by $1/\|r\|_2^2$. Figure 3a shows that neither change alone helps consistently: energy geometry weighted by $1/\|r\|_2^2$ is worse than Residual-DCDM on both PDEs, and inverse-Z weighting of the residual loss helps Darcy but hurts screened Poisson. Only the combination, E-DCDM, helps on both. The combined gain exceeds what the two separate changes predict: computed within each seed, the multiplicative field-error interaction is 0.100–0.107 on Darcy and 0.064–0.072 on screened Poisson, where values below one indicate a combination-specific gain. The proposed method is therefore the complete loss, not energy geometry under arbitrary weighting; Appendix A.5 gives the cell values, per-seed checks, and energy and residual errors.

Removing the overshoot penalty closes a setting-dependent share. A post hoc step-overshoot control (M) trains the residual loss without its overshoot penalty, scoring each direction by its minimal-residual step while deploying, like every model here, the shared energy step. Its share of the log field-error gap between Residual-DCDM (R) and E-DCDM (E), $f = \log(R/M)/\log(R/E)$, is 0.03–0.18 on screened Poisson under the shared recipe, 0.12–0.78 with each loss tuned (Figure 3b), and 0.34–0.45 on Darcy (shared recipe); E-DCDM beats the control on every seed, and on screened Poisson at all nine configurations (Table 10). At the deployed energy step the control's directions still carry 1.2–$1.7\times$ Residual-DCDM's overshoot penalty on screened Poisson (3.4–$3.9\times$ on Darcy), and the control lacks the inverse-Z weighting (Appendix A.5).

The coarse normalizer is consequential. The energy-loss groups of Figure 3a replace Z_{32} inside the same energy loss. Every tested 2D coarse Galerkin normalizer (8, 32, or 64 sine modes, or 32 local hats) beats the same-seed Residual-DCDM on every seed of both PDEs, by at least $4.59\times$ in field error (a post hoc summary), whereas the generic Z_{SD} is worse than Residual-DCDM on every Darcy seed and mixed on screened Poisson. In a post hoc control, the solution-supervised exact normalizer $r^\top A^{-1} r$ beats Residual-DCDM 4.0–$5.6\times$ but trails Z_{32} on every seed: by 1.75–$2.38\times$ on Darcy (1.48–$2.34\times$ with the transferred settings; Darcy was not tuned) and, with each loss tuned, 1.32–$1.80\times$ on screened Poisson (Figure 3b), where at the same unclipped configuration it is still

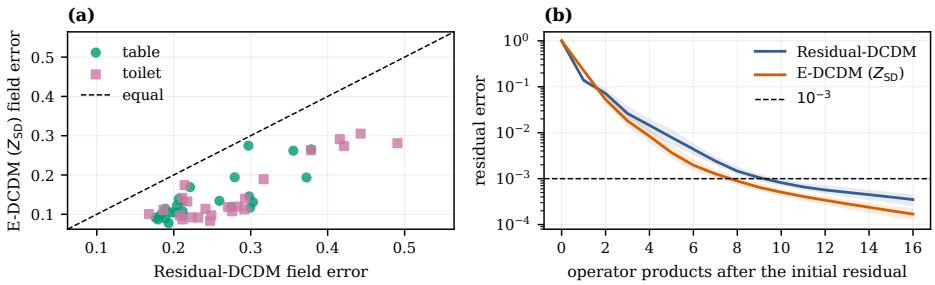

Figure 4: E-DCDM (Z_{SD}) improves every one of the 47 held-out 3D operators and reaches relative residual 10^{-3} earlier. (a) Each point averages five paired seeds and 20 shared states; all lie below equality. (b) Median rollout with interquartile bands over 470 runs across seeds and problems.

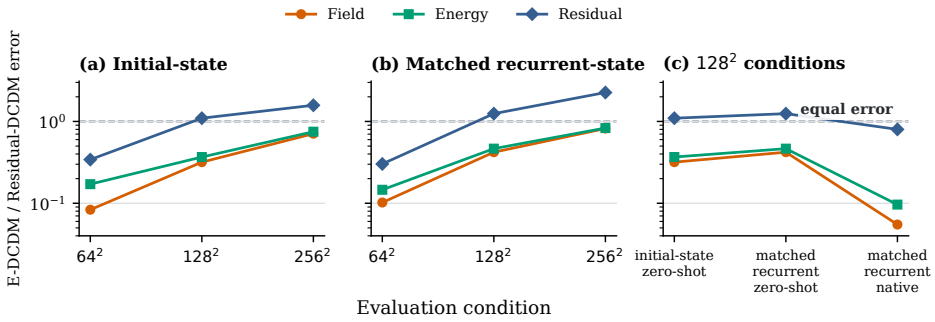

Figure 5: Zero-shot refinement preserves field and energy gains but reverses the residual ranking. Eight-product error ratios, E-DCDM/Residual-DCDM (below one favors E-DCDM). **(a, b)** The reversal occurs for initial-state and matched recurrent-state 64^2 checkpoints. **(c)** At 128^2 it persists for the matched 64^2 checkpoint and vanishes when training at 128^2.

1.32–1.68$\times$ worse. Under matched states the coarse weighting therefore does more than approximate relative energy; with warm starts alone the ranking reverses on screened Poisson (Appendix A.5). The coarse resolution still matters. Relative to the same-seed Z_{32} model, Z_8 lowers field error on every seed of both PDEs, whereas Z_{64} is worse on every Darcy seed; matching the training mean of the nominal inverse weights preserves these conclusions, and the equally sized local-hat basis is not equivalent to sine-32 (Appendix A.6). We keep Z_{32} as the primary configuration, since Z_8 was identified only in this comparison.

5.3 Where the gain holds: 3D operators, resolution, and robustness

Independent 3D operators. Both primary 3D studies use the energy loss with the generic normalizer Z_{SD} rather than the 2D sine basis, so they test energy decrease under generic weighting, not transfer of Z_{32}; in 2D, Z_{SD} did not reproduce the Z_{32} gain (Section 5.2). On 47 ModelNet10-derived operators from unseen categories, energy training lowers category-balanced one-step field error $1.8\times$ (operator-bootstrap 95% interval $[1.7, 1.9]\times$); every operator and all five paired seeds improve, and it reaches residual tolerance in fewer operator products (Figure 4; counts exclude inference cost), although its one-step residual error is $1.38\times$ higher. An exploratory follow-up, specified after these results with favorable-only reporting (outputs released regardless), uses a Galerkin normalizer on 64 local hats and lowers one-step field error a further 7% (ratio 0.930, $[0.908, 0.951]$; 41 of 47 operators), so coarse weighting is incremental here. A shared-operator 64^3 pressure test also improves field error and time to tolerance but worsens the immediate residual on every seed (Appendix A.3).

Zero-shot resolution transfer exposes a metric boundary. Fixed 64^2 checkpoints, initial-state or matched, retain lower field and energy error at 128^2 and 256^2 but have higher residual error (Figure 5); absolute errors rise sharply, so this is ranking transfer, not resolution invariance. A 256^2 frequency audit agrees: E-DCDM removes more low-frequency field error but leaves more high-frequency residual. Training at 128^2 makes E-DCDM better on all three metrics, but it also changes state coverage (Appendix A.7).

Contrast and other checks. Without retraining (initial-state checkpoints), E-DCDM remains favorable when Darcy contrast doubles to eight, but both learned solvers fail at 100:1 contrast, with energy training worse. Further checks are in Appendices A.5–A.8.

5.4 Cost and access boundary

The learned methods share one CNN call, cached orthogonalization, and one operator product per step. On a fixed 32-case Darcy batch, the two learned solvers bring half of the cases to relative residual tolerances of 10^{-2} and 10^{-3} within 4 to 6 operator products after the initial residual, whereas CG and Jacobi-PCG need 94 to 180 (Appendix A.8, Table 22). At 10^{-3} both learned solvers need six products, and in repeated, randomly interleaved timing on one GPU their runtimes agree within 0.3% (0.735 versus 0.737 ms per case). At 10^{-2}, E-DCDM needs 4 rather than 5 products and 20% less time (0.486 versus 0.610 ms). In floating-point operations one 2D network call equals about 1.5×10^4 operator products, so learned solves do far more arithmetic; their lower GPU times reflect CG's launch-bound small products (Table 20). A post hoc 32-product extension favored E-DCDM at every tolerance in both families, by a full product per seed only on Darcy (Appendix A.8).

AMG is outside the product-count comparison because a V-cycle is not one Av and its timing uses a different CPU access path. AMG can ultimately reach greater accuracy and is the appropriate choice when explicit stencil access and repeated solves amortize its reusable hierarchy (Appendix A.8). Because the network reads the coefficient field, which here determines the stencil, the product budget does not exclude assembled deflation or rediscretized multigrid; it budgets opaque operators, not work. Arbitrary learned directions also require a classical fallback.

6 Discussion and limitations

When to use the energy objective. The choice should follow the quantity that matters downstream. If field or energy accuracy is the target and slowly varying error is costly, both 2D families and the 3D operators (with Z_{SD}) favor energy training; if the stopping criterion is residual norm, especially under a resolution shift, residual training can remove high-frequency residual faster (Appendix A.7). E-DCDM needs only a differentiable product Ad and a positive-definite energy. The most transferable lesson is diagnostic: report field, energy, and residual error together, match the recurrent-state distribution, and inspect which spectral components each correction removes.

Limitations. We test structured-grid 2D systems, one voxelized pressure benchmark whose right-hand sides share two operators, and one synthetic 32^3 elliptic family (two unseen categories, one contrast and architecture, manufactured forcing; the bootstrap conditions on those operators); unstructured, nonsymmetric, nonlinear, and industrial-scale systems remain untested. The sine basis of Z_{32} needs a rectangular grid; local hats could serve unstructured meshes but are untested there. The 3D studies use Z_{SD}, so they do not show that the primary weighting transfers, and why Z_{SD} helps in 3D but not in 2D is open. The size and family of the coarse space change the result, and we have no validated rule for choosing them; selecting the size by field or energy error on development problems is a natural but untested procedure. Clipping is active in most shared-recipe batches (Table 14), separate tuning covers screened Poisson only, the step-overshoot control's arms clip different fractions of batches, and we do not test a conjugate-residual (GCR-type) learned solver. The main method, the matched comparison, and most controls (including the optimizer, exact-normalizer, and step-overshoot controls) are post hoc; the resolution, FNO, and 3D studies, though also post hoc to the primary study, fixed their analyses before scoring (Appendix A.2, Protocol chronology). The matched training set is fixed, so success beyond it is unproved. The gains are relative and occur at already small absolute errors (screened Poisson field error 6.9×10^{-10} versus 6.4×10^{-9}), and timings use one fixed 32-case batch.

Conclusion. Changing only the training loss of a learned conjugate-direction solver, from residual decrease to coarse-normalized exact energy decrease, lowers field error 6.6–$9.8\times$ across paired seeds under matched states and a shared recipe (3.3–$5.4\times$ on screened Poisson with separate tuning). Energy decrease shifts which spectral errors are removed and the state weighting sets how training states compete; in 2D neither alone suffices, and under matched states the coarse weighting beats even the exact energy normalizer. With a generic normalizer the one-step gain extends to held-out 3D operators; residual error can be higher, and both learned solvers fail at extreme contrast.

AI USE STATEMENT

We used generative AI tools to help develop theoretical and conceptual ideas, formulate and assist with mathematical claims and proofs, refine hypotheses, design experiments, implement methods, generate synthetic 3D operators, prepare public datasets, and interpret results. Translation and qualitative or thematic data analysis were not applicable. We also used generative AI for brainstorming and literature discovery, suggesting experimental parameters, writing and editing code, creating and refining scientific figures (including the schematic layout of Figure 1; its heatmaps are real model outputs), drafting and editing text, suggesting paper structure, and formatting references. We reviewed all AI-assisted work. Key mathematical identities were numerically verified in the released test suite; reported results are reproducible from released per-case outputs or checked against hashed sealed outputs; protocol records document which analyses were fixed before scoring versus post hoc; and citations were checked against primary sources. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work uses only public numerical datasets and synthetic operators derived from public geometry; it involves no human subjects, personal data, or sensitive content. Its contribution is a training objective for general-purpose linear solvers, and we are not aware of specific harmful applications beyond those of numerical simulation in general. Post hoc analyses and the one conditionally reported follow-up are identified in Appendix A.2, and their outputs are released regardless of outcome.

REPRODUCIBILITY STATEMENT

The anonymous repository (`https://anonymous.4open.science/r/E-DCDM-1C28`) releases the implementation, the frozen training and evaluation sources, the evaluated checkpoints of the reported comparisons (including the factorial, normalizer-size, and basis-family studies), per-case arrays, protocol records, and replay and figure scripts, with data splits, environments, seeds, and operator-product accounting. Most aggregates replay from released per-case data; the factorial cell means and intervals, the eigenmode summaries, the basis-family intervals, and some control summaries are compact copies of sealed outputs with their hashes. Cached training histories, upstream warm-start models, and the initial-state Residual-DCDM checkpoints other than the selected Darcy model are not included, so the released checkpoints can be re-evaluated and the reported aggregates replayed or hash-checked, but the models cannot be retrained end to end. For the optimizer, exact-normalizer, and step-overshoot controls, the repository releases code, protocols, and aggregates with sealed hashes, but not checkpoints, exact-normalizer labels, or per-case arrays.

REFERENCES

Heming Bai and Xin Bian. Hybrid deep learning and iterative methods for accelerated solutions of viscous incompressible flow. *Journal of Computational Physics*, 554:114747, 2026. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2026.114747. URL https://www.sciencedirect.com/science/article/pii/S0021999126000975.

Lianghao Cao, Thomas O'Leary-Roseberry, Prashant K. Jha, J. Tinsley Oden, and Omar Ghattas. Residual-based error correction for neural operator accelerated infinite-dimensional bayesian inverse problems. *Journal of Computational Physics*, 486:112104, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2023.112104. URL https://www.sciencedirect.com/science/article/pii/S0021999123001997.

Nunzio Dimola, Alessandro Coclite, and Paolo Zunino. Neural preconditioning via krylov subspace geometry. *ArXiv*, abs/2507.15452, 2025. URL https://api.semanticscholar.org/CorpusID:280284246.

Mohammad Sadegh Eshaghi, Cosmin Anitescu, Navid Valizadeh, Yizheng Wang, Xiaoying Zhuang, and Timon Rabczuk. Nows: Neural operator warm starts for accelerating iterative solvers. *ArXiv*, abs/2511.02481, 2025. URL https://api.semanticscholar.org/CorpusID:282748964.

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.

Julius Herb and Felix Fritzen. Accelerating conjugate gradient solvers for homogenization problems with unitary neural operators. *International Journal for Numerical Methods in Engineering*, 127 (5):e70277, 2026. doi: https://doi.org/10.1002/nme.70277. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.70277.

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49:409–435, 1952. URL https://api.semanticscholar.org/CorpusID:2207234.

Jun-Ting Hsieh, Shengjia Zhao, Stephan Eismann, Lucia Mirabella, and Stefano Ermon. Learning neural PDE solvers with convergence guarantees. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rklaWn0qK7.

Prashant K. Jha. Residual-based error corrector operator to enhance accuracy and reliability of neural operator surrogates of nonlinear variational boundary-value problems. *Computer Methods in Applied Mechanics and Engineering*, 419:116595, 2024. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2023.116595. URL https://www.sciencedirect.com/science/article/pii/S0045782523007193.

Haina Jiang, Liam Wang, Peng-Chen Chen, Min Seop Kwak, Seungryong Kim, Brian Bell, and Jeong Joon Park. Error-conditioned neural solvers, 2026. URL https://arxiv.org/abs/2606.27354.

Ayano Kaneda, Osman Akar, Jingyu Chen, Victoria Kala, David Hyde, and Joseph Teran. A deep conjugate direction method for iteratively solving linear systems. In *International Conference on Machine Learning*, 2022. URL https://api.semanticscholar.org/CorpusID:252683427.

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://doi.org/10.1137/24M162861X.

Nikola B. Kovachki, Zong-Yi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *J. Mach. Learn. Res.*, 24:89:1–89:97, 2023. URL https://api.semanticscholar.org/CorpusID:259149906.

Yichen Li, Peter Yichen Chen, Tao Du, and Wojciech Matusik. Learning preconditioner for conjugate gradient pde solvers. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:258947763.

Zhihao Li, Di Xiao, Zhilu Lai, and Wei Wang. Neural preconditioning operator for efficient pde solves. *ArXiv*, abs/2502.01337, 2025. URL https://api.semanticscholar.org/CorpusID:276107576.

Zong-Yi Li, Nikola B. Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *ArXiv*, abs/2010.08895, 2020. URL https://api.semanticscholar.org/CorpusID:224705257.

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. *ArXiv*, abs/2003.05744, 2020. URL https://api.semanticscholar.org/CorpusID:212675750.

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*, 24:355–365, 1987. URL https://api.semanticscholar.org/CorpusID:121660604.

Yvan Notay. Flexible conjugate gradients. *SIAM Journal on Scientific Computing*, 22(4):1444–1460, 2000. doi: 10.1137/S1064827599362314. URL https://doi.org/10.1137/S1064827599362314.

Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bezenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=MtekhXRP4h.

Alexander Rudikov, Vladimir Fanaskov, Ekaterina Muravleva, Yuri M. Laevsky, and Ivan Oseledets. Neural operators meet conjugate gradients: The FCG-NO method for efficient PDE solving. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42766–42782. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/rudikov24a.html.

Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h. A deflated version of the conjugate gradient algorithm. *SIAM Journal on Scientific Computing*, 21(5):1909–1926, 2000. doi: 10.1137/S1064829598339761. URL https://doi.org/10.1137/S1064829598339761.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003. doi: 10.1137/1.9780898718003. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898718003.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *ArXiv*, abs/2210.07182, 2022. URL https://api.semanticscholar.org/CorpusID:252873473.

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.

Zengkai Wu, Sheng Sun, Min Tang, and Ping Li. A fourier neural operator enhanced physics-embedded iterative learning solver for electromagnetic scattering analysis. *IEEE Antennas and Wireless Propagation Letters*, 24:1954–1958, 2025. URL https://api.semanticscholar.org/CorpusID:277136290.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, 2015. URL https://api.semanticscholar.org/CorpusID:206592833.

Yu Xing, Yang Liu, Lipeng Chen, Huiping Tang, and Lin Lu. Pcg-informed neural solvers for high-resolution homogenization of periodic microstructures, 2025. URL https://arxiv.org/abs/2506.17087.

Enrui Zhang, Adar Kahana, Alena Kopaničáková, Eli Turkel, Rishikesh Ranade, Jay Pathak, and George Em Karniadakis. Blending neural operators and relaxation methods in pde numerical solvers. *Nature Machine Intelligence*, 6:1303 – 1313, 2022. URL https://api.semanticscholar.org/CorpusID:251903572.

Haolan Zheng, Yanlai Chen, Jie-Qun Han, and Yue Yu. Rebano: Reduced basis neural operator mitigating generalization gaps and achieving discretization invariance. *ArXiv*, abs/2509.09611, 2025. URL https://api.semanticscholar.org/CorpusID:281252233.

A Reproducibility details and additional controls

Appendix A.1 gives models, training, cost accounting, and the projection; Appendix A.2 the estimand, uncertainty, and protocol chronology; Appendix A.3 the 3D construction and the shared-operator pressure audit; Appendix A.4 per-seed and trajectory audits; Appendix A.5 the mechanism controls (eigenmodes, factorial, supervised reference); Appendix A.6 the normalizer sensitivity; Appendix A.7 transfer and failure boundaries; and Appendix A.8 robustness checks and classical baselines, including the learning-rate check.

A.1 Models, training, and operator-product accounting

Both primary networks receive two 62×62 fields: residual divided by its root-mean-square magnitude and the centered log coefficient. A 28-channel full-resolution stream, one half-resolution stream, skip connections, and a one-channel decoder produce a dense direction. Each graph has 99,737 parameters. The output scale is immaterial because the exact line search sets its deployed magnitude. The Darcy operator uses harmonic face fluxes and has 62^2 interior unknowns; this discretization yields an SPD linear system. Both losses train from scratch on the same matched recurrent-state training set and select checkpoints by their own held-out objectives.

Table 1: The primary recurrent comparison changes only the complete training loss. A checkmark marks a shared component; each loss selects its own checkpoint using held-out validation states.

Component	Residual-DCDM	E-DCDM
CNN / trainable parameters	dense / 99,737	same ✓
Training states	shared depths 0–6	same ✓
Darcy seeds; screened seeds	811–813; 821–823	paired ✓
Optimizer / learning rate	Adam / 10^{-4}	same ✓
Batch / epochs / clipping	32 / 1,000 / 1	same ✓
Cached-span projection	yes	same ✓
Held-out selection	residual loss	energy loss
Online recurrence / operator products	seven full-memory / eight	same ✓
Solution labels	none	none ✓

Table 2: Matched recurrent-state training lowers both field-error endpoints while preserving the advantage of the complete E-DCDM loss. Exact relative field errors after eight operator products correspond to Figure 2a; validation selects seed 812 for both Darcy arms and residual seed 822 and energy seed 823 on screened Poisson.

Problem	Training states	Residual-DCDM	E-DCDM
Darcy	initial state	$6.251e{-5}$	$5.210e{-6}$
Darcy	matched recurrent state	$6.432e{-6}$	$6.569e{-7}$
Screened Poisson	initial state	$1.008e{-5}$	$6.297e{-8}$
Screened Poisson	matched recurrent state	$6.431e{-9}$	$6.875e{-10}$

Residual-DCDM and E-DCDM are trained with Adam at 10^{-4}, batch size 32, gradient clipping at one, and 1,000 deterministic epochs. They start from paired initial weights for seeds 811–813 on Darcy and 821–823 on screened Poisson. Under recurrent training, validation selects seed 812 for both Darcy objectives, residual seed 822 and energy seed 823 on screened Poisson. Checkpoint selection minimizes each model's own balanced objective. Darcy has 160 training and 40 validation coefficient fields. The shared fixed training set samples residual- and energy-source trajectories at depths 0–6, giving 2,560 training and 640 validation states; screened Poisson has 576 and 192. The initial-state arm uses the earlier 1,280/320 Darcy training set. No test instance enters either training or selection set.

Every recurrent training state stores the preceding directions and operator images. Backpropagation traverses the cached-span projection and one fresh application of A to p_ψ, but never a numerical solution. The fixed training set is shared byte-for-byte across objectives and is not regenerated from the model being trained. Both methods then use the same full-memory recurrence at deployment.

The released receipts record 35.0 to 35.5 minutes per Darcy energy run and 8.1 to 8.9 minutes per screened-Poisson energy run; matched residual runs take 37.2 to 39.9 and 8.0 to 8.6 minutes, respectively. These are observed run times, not cross-hardware benchmarks. Each 1,000-epoch run evaluates one differentiable operator product per training state and epoch, or 2.56 million Darcy and 0.576 million screened-Poisson operator products, plus validation. A checkpoint occupies 408,641 bytes in the primary 64-grid release. The cached shared training sets occupy 308 and 74 MiB compressed. Creating each training set advances two fixed source trajectories for six corrections, which costs 12 operator products per initial state: 19,200 for Darcy and 4,608 for screened Poisson.

Table 3: Three ways to normalize the energy decrease; only the first two are solution-free. Sensitivity refers to the 2D normalizer study in Section 5.2.

Normalizer	Definition	Extra offline cost	Used for
Coarse Z_N	$g^\top G^{-1} g$ (N vectors)	N images Av_j per operator	2D primary ($N = 32$); sensitivity
Generic Z_{SD}	$(r^\top r)^2/(r^\top Ar)$	at most one Ar per state	3D studies; 2D sensitivity
Exact	$e^\top Ae = r^\top A^{-1} r$	requires the solution	supervised references

The operator-product count includes the initial residual $r_0 = b - Ax_0$. Each DCDM direction consumes one additional application of A. Full-memory orthogonalization reuses cached Ap_j and therefore changes storage and vector arithmetic, but not this logical operator-product count. The exact black-box 32-mode reference needs $1 + 6 + 32 = 39$ operator products: r_0, six DCDM directions, and the 32 images Av_j; its final update reuses those images. In method names, a suffix k denotes k learned corrections, and FM denotes full-memory orthogonalization.

Operator access and cost accounting. Network inference, vector operations, and inner products are timed but not counted as operator products. Offline products (Table 3) are reported rather than hidden in the deployment budget; the N images Av_j behind Z_N are reused across an operator's training states, and our structured-grid code forms the same Gram matrix from the stencil. Counting operator products models callable operators whose coefficients are visible to a learned component but whose assembly is unavailable online. Because the network reads the coefficient field, which on our structured grids determines the stencil, this setting does not exclude rediscretized multigrid or assembled deflation; operator products are therefore a budget for the opaque-operator setting only, not a measure of arithmetic cost. Appendix A.8 measures direct coarse assembly, AMG, and the arithmetic cost of the network.

Cached-span projection. At step t, let $P = [p_0, \ldots, p_{t-1}]$ and $Q = AP$. The new raw direction's cached-span component is removed by

$$c = (P^\top Q)^{-1} Q^\top d, \qquad p = d - Pc. \tag{8}$$

This is classical full-memory conjugate-direction orthogonalization, identical for the two learned losses; it uses cached products and no new call to A. Because the cached directions are A-conjugate, $P^\top AP$ is diagonal, so the evaluator applies Equation 8 sequentially, $p \leftarrow p - p_j\,(p^\top Ap_j)/(p_j^\top Ap_j)$ for each cached j, reusing the cached Ap_j rather than solving a potentially ill-conditioned Gram system. When $P^\top r = 0$, projection preserves $p^\top r = d^\top r$ and reduces the denominator to $p^\top Ap = d^\top Ad - c^\top (P^\top AP)c$. Hence it cannot reduce Δ_A. Any exact line step with $p^\top Ap > 0$ already cannot increase energy error; the projection's content is that the step is optimal over $\mathrm{span}(P, d)$ rather than over d alone. The guarantee concerns energy, not field error: eigenvalue bounds give $\lambda_{\min}\|e\|_2^2 \le \|e\|_A^2 \le \lambda_{\max}\|e\|_2^2$, so energy monotonicity controls L^2 only up to the condition number and does not imply per-step L^2 monotonicity. This is why the headline L^2 claim is empirical and paired, while this projection fact only explains the shared recurrence.

For $w = VG^{-1}g$, Galerkin orthogonality gives $V^\top A(e - w) = 0$ and therefore

$$r^\top A^{-1} r = \|e\|_A^2 = \|w\|_A^2 + \|e - w\|_A^2, \qquad \|w\|_A^2 = Z.$$

Thus $Z \leq r^\top A^{-1} r$, with strict positivity when $g \neq 0$. The matched recurrent trainer evaluates the loss using $\max(Z, 10^{-20})$, with Z computed on the RMS-normalized residual; the training set generator rejects $Z \leq 10^{-18}$ before training. No stored state activates either threshold. Because Δ_A can exceed Z, the resulting loss is not restricted to $[0, 1]$.

The 32-mode normalizer Z in Equation 7 is recomputed for each training state from a Gram matrix formed once per operator. Being constant in the network, it leaves each state's optimum unchanged but alters cross-state weighting for a finite-capacity shared network.

Primary recurrence arithmetic, operator products, line searches, and reported metrics use float64; CNN inference uses float32 before casting its direction to float64. The evaluator rejects nonfinite directions and nonpositive curvature $p^\top A p \leq 10^{-24}$. Reference targets are accepted only after a true relative residual below 10^{-11} (and below 10^{-12} for the supervised-label construction).

A.2 Estimand and uncertainty

A warm-start model is identified by its architecture, independent training-data root, and ensemble member. The Darcy population has ten FNO models (five members for each of roots B and C) and six U-Net models (three members for each root). A *warm-start cell* is one combination of architecture and data root. For each coefficient field, we first average members within its warm-start cell and then average the four warm-start cell means with equal weight. Thus the primary sample size for field uncertainty is 1,000, not 16,000. A shared bootstrap draw resamples these coefficient instances and applies the same indices to every method and warm-start model. Reported paired intervals are the 2.5 and 97.5 percentiles over 20,000 draws. They do not include checkpoint-retraining uncertainty.

Primary paired estimates. Relative to initial-state training, the matched recurrent-state Residual-DCDM field endpoint is $9.7\times$ lower on Darcy and more than three orders of magnitude lower on screened Poisson; the E-DCDM endpoint also improves. After eight operator products, all 1,000 Darcy instances, 16 warm-start model means, four balanced warm-start cells, 96 screened coefficient identities (each evaluated in all four strata), and four screened strata favor E-DCDM. The paired field-difference intervals are $[5.649, 5.903] \times 10^{-6}$ on Darcy and $[4.994, 6.561] \times 10^{-9}$ on screened Poisson. Exact endpoints appear in Table 2.

Protocol chronology. Darcy development fields come from `darcy_train_64.pt` (Zenodo 12784353) and evaluation fields from `piececonst_r421_N1024_smooth2.mat` (Zenodo 12529654; Section 5). The 128^2 and 256^2 transfers instead use fields 0–255 and 256–511 of `darcy_test_64.pt` (Zenodo 12784353), upsampled by nearest neighbor; neither set overlaps the training split. The warm-start models were trained on separate, non-overlapping index ranges of `darcy_train_64.pt`. The Darcy and screened-Poisson evaluation populations had been scored by earlier residual-trained and exploratory methods before the energy objective was introduced, and an earlier 500-epoch version of the method was scored on them before the reported recipe; the main method and the matched-state comparison are therefore post hoc. Checkpoints are always selected on validation data. The 32 sine modes and the eight-product budget were inherited from earlier development work, and the learning rate 10^{-4} and 1,000 epochs follow the released DCDM defaults.

The resolution-transfer, native-128^2 matched recurrent-state, FNO, shared-operator 64^3 pressure, and ModelNet10-derived 3D studies each fixed their data, selection rule, metrics, and analysis before the corresponding outputs were scored. The anonymous repository retains the dated specifications, code hashes, implementation corrections, and complete outputs. These records establish the order of analysis decisions but are not third-party preregistrations.

The following were added after the primary study: the objective-by-weighting factorial (screened Poisson assembles cells whose training and evaluation were fixed beforehand; the Darcy arms are a later replication on fixed selected checkpoints), the normalizer-size, generic-scale, and local-hat sensitivity studies, and the summary that every tested 2D coarse normalizer stays at most 0.22 times the same-seed Residual-DCDM field error. That summary is a check on already reported results, and its threshold was chosen after a rough estimate was known. The 3D coarse-normalizer follow-up (Appendix A.3) was specified after the Z_{SD} 3D results were known but before its normalizer was built or any arm trained; its protocol fixed the basis, arms, endpoint, interpretation branches, and the rule that it would enter this paper only if favorable. Its outputs are released regardless. The optimizer

and exact-normalizer controls, and their extension to a nine-configuration grid with the tuned exact normalizer and the step-overshoot control (Appendices A.5 and A.8), each followed a protocol that fixed arms, selection rules, interpretation branches, sealed predictions, and hashed inputs before training; they are post hoc on the opened evaluation sets, and every predeclared outcome is reported, including the sealed predictions that failed.

A.3 3D DETAILS: MODELNET10 AND SHARED-OPERATOR PRESSURE

The official ModelNet10 archive has SHA-256 `9d8679435fc07d1d26f13009878db164a7aa8ea5e7ea3c8880e42794b7307d51`. Lexicographic selection gives 72 training, 16 validation, and 48 test meshes; two test meshes voxelize identically, so claims use 47 unique operators (24 table and 23 toilet). Each operator contributes two right-hand sides, two state sources, and depths 0–4. Within an operator, these 20 states are averaged; operators are averaged within category and the two categories receive equal weight. The bootstrap resamples unique operators within category. No seed is selected: each operator first averages all five separately validation-selected models. The protocol was frozen before state preparation; a pre-test amendment clarified seed averaging, category balance, duplicate handling, and the exact bootstrap estimand before any test label was opened. This study normalizes energy decrease by $Z_{\mathrm{SD}} = (r^\top r)^2/(r^\top A r)$ rather than importing the 2D sine basis. The shared-operator 64^3 extension makes the same choice; a coarse alternative is tested below. Consequently, the 3D results support the complete energy-decrease objective under this generic solution-free scale, not energy geometry in isolation; the stronger primary 2D endpoint additionally depends on its tested coarse state weighting.

Table 4: Every paired seed favors E-DCDM (Z_{SD}) on independent 3D operators. Field entries are category-balanced one-step errors.

Seed	Residual-DCDM	E-DCDM (Z_{SD})	Reduction
1201	.2417	.1787	26.1%
1202	.2456	.1381	43.8%
1203	.2696	.1436	46.7%
1204	.3153	.1350	57.2%
1205	.2471	.1389	43.8%

Averaging the five seeds gives category-balanced field error 0.264 for Residual-DCDM and 0.147 for E-DCDM (Z_{SD}). The reductions are 42.8% for table and 45.7% for toilet. Within 16 operator products after the initial residual, E-DCDM (Z_{SD}) reaches relative residual 10^{-3} on all 470 runs across seeds and problems, with median eight products, versus 465/470 and median ten for Residual-DCDM. These product counts exclude inference and history-management cost.

Coarse normalizer on the 3D operators (follow-up). To test whether coarse state weighting also helps in 3D, we replaced Z_{SD} by a Galerkin normalizer $Z_{64}^{\mathrm{hat}} = g^\top G^{-1} g$, $G = V^\top A V$, where V is the $4 \times 4 \times 4$ tensor product of the one-dimensional Dirichlet piecewise-linear hats of the 2D basis-family control (Appendix A.6), each unit-normalized (rank 64). Hats rather than sines were fixed in advance because the manufactured 3D solutions are sums of sine modes of frequency 1–4, which a sine coarse space would nearly contain. Everything else is unchanged: states, network, optimizer, epoch cap, batch order, and per-seed initial weights; the Residual-DCDM and Z_{SD} checkpoints are reused. A 20-epoch replay of the Z_{SD} run through the new trainer reproduced its validation curve exactly, as did re-evaluation of the reused arms. The primary arm uses raw Z_{64}^{hat}; a secondary arm rescales it to the training mean of $1/Z_{\mathrm{SD}}$. On depth-0 training states, Z_{64}^{hat} captures a median 30% (maximum 59%) of the exact error energy.

Raw Z_{64}^{hat} lowers all three one-step errors relative to Z_{SD} (field 0.930, energy 0.959, residual 0.956), in both categories (table 0.909, toilet 0.950), on 41 of 47 operators and all five seeds. Against Residual-DCDM it lowers field error by 48.2% ($[44.4, 52.2]\%$; 44.3% for Z_{SD}) with every operator improving, and it meets all five predeclared usefulness criteria, but its one-step residual error remains 1.32 times higher (1.38 for Z_{SD}). The gain is modest and one-step: rollouts reach relative residual 10^{-2} and 10^{-3} on all 470 runs with the same medians as Z_{SD} (4 and 8 products), and median field error is 7% lower after seven corrections but 2% higher after 16. The mean-matched arm has a similar pooled ratio but favors hats on only three seeds and 38 operators, and 99.6–100% of training batches

Table 5: A local-hat coarse normalizer modestly improves on Z_{SD} in 3D. Category-balanced one-step field error; ratios are relative to same-seed Z_{SD} (below one favors hats). Residual-DCDM: .2639.

Seed	Z_{SD}	Hat raw	Ratio	Hat mean-matched	Ratio
1201	.1787	.1520	.851	.1317	.737
1202	.1381	.1363	.987	.1361	.986
1203	.1436	.1359	.947	.1496	1.042
1204	.1350	.1295	.959	.1352	1.002
1205	.1389	.1294	.932	.1325	.954
Pooled	.1469	.1366	.930	.1370	.933
95% interval			[.908, .951]		[.909, .954]

clip in every arm audited, so, as in 2D, this does not isolate basis geometry from optimization under the fixed clipping rule. The follow-up therefore shows that a preselected coarse Galerkin normalizer is compatible with the 3D result and improves it modestly; it does not show that sine Z_{32} transfers, or that coarse weighting is needed in 3D, where Z_{SD} already provides most of the gain.

Shared-operator 64^3 pressure paired-seed audit. The shared-operator 64^3 pressure study changes application and scale but shares fixed matrices within its two simulations. The validation-selected E-DCDM (Z_{SD}) model lowers field error by 59.4% on all six held-out right-hand-side frames (84 states: 14 source-by-depth roles per frame) and reaches relative residual 10^{-3} in 9–10 operator products after the initial residual; matched Residual-DCDM reaches none within 16. We therefore treat this result as descriptive support rather than six independent replicates (a reading adopted after scoring).

Table 6: Every shared-operator 64^3 pressure seed improves field and energy error. Entries are Residual-DCDM / E-DCDM (Z_{SD}) means across all 84 held-out states; validation, not these errors, selects seed 852 for both arms.

Seed	Field error	Energy error	Residual error
851	.673/.277	.318/.142	.0950/.145
852	.732/.298	.326/.143	.0953/.139
853	.551/.343	.301/.157	.0923/.147

All paired seeds improve field and energy error but worsen the immediate residual, consistent with the objective weighting. Full 16-step trajectories for every seed and all four baselines are released in the anonymous repository.

A.4 Paired-seed and native-grid trajectory audits

Table 7: Every paired seed favors energy training on field error. Entries are reductions of the balanced mean error from residual- to energy-objective training after eight operator products; field error improves for every instance, warm-start model, and stratum.

Problem	Paired seed	Field	Energy	Residual
Darcy	811	88.0%	83.5%	65.0%
	812	89.8%	85.4%	69.8%
	813	89.6%	85.6%	67.9%
Screened	821	84.8%	79.7%	56.9%
	822	85.1%	80.0%	57.2%
	823	88.8%	84.5%	66.5%

This audit pairs networks initialized from the same weights and evaluates all six pairs; no seed is selected from test outcomes. It is a finite sensitivity check, not a confidence interval over optimization randomness.

FNO direction network. The architecture control replaces the CNN by the official NeuralOperator FNO (four layers, width 16, 12 modes per direction; 89,553 parameters versus 99,737) behind the same input channels, losses, line step, projection, and recurrence. It trains on the same matched recurrent Darcy states (2,560 training and 640 validation rows) with paired seeds 841–843, Adam at 10^{-4}, clipping at one, and 1,000 epochs, and each loss selects its own checkpoint by validation. On the 1,000 evaluation fields crossed with all 16 warm-start models, E-DCDM lowers eight-product field error from 9.38×10^{-2} to 1.28×10^{-3} (98.6%), energy error by 95.8%, and residual error by 89.0%; every instance, warm-start model, and cell favors it. Both FNO arms stay far above the CNN errors, and the residual-trained FNO is nearly flat after its first correction. The analysis was fixed before any FNO checkpoint was trained or evaluated, but the control is post hoc to the primary study.

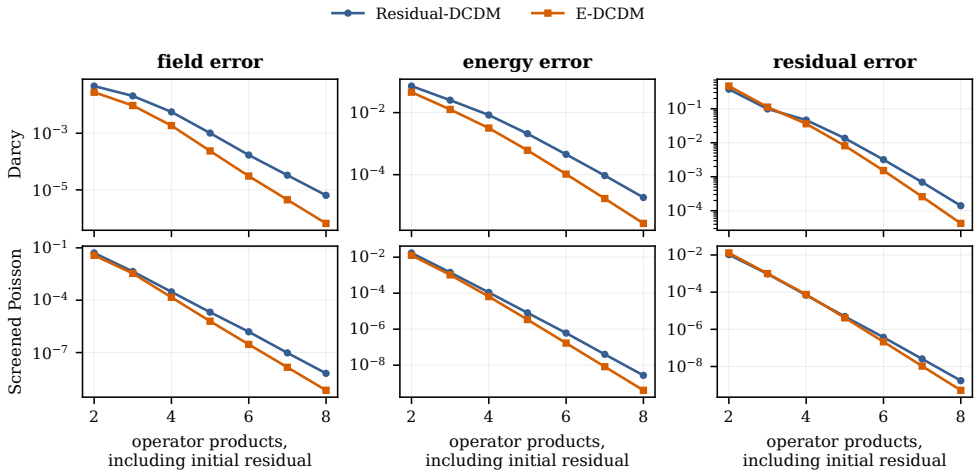

Figure 6: All three errors fall at every operator product under matched recurrent-state training. E-DCDM is lower in field and energy error throughout and in residual error from product four (Darcy) or five (screened Poisson) onward. Curves use validation-selected checkpoints.

After eight operator products, E-DCDM lowers Darcy energy and residual errors by 85.4% and 69.8%, and screened-Poisson energy and residual errors by 85.2% and 69.8%, respectively.

A.5 Mechanism controls

Expanded verified numerical eigenmode audit. The main spectral result uses all 640 states in the matched recurrent-state Darcy validation set, spanning 40 coefficient fields and 28 strata defined by architecture, source, and depth. We cache 16 eigenvectors at either end of each operator's spectrum, balance the strata, and bootstrap coefficient instances so repeated states from one operator remain clustered. The maximum relative eigenpair residual is 9.99×10^{-11}. Under the same additional inverse-Z weighting and initialization seed matched in three pairs, energy geometry removes 4.02 percentage points more low-mode field power, with 95% interval $[3.52, 4.54]$, and wins all 28 strata for every seed. Residual geometry removes 14.84 points more high-mode residual power, with interval $[10.46, 19.46]$; after seed averaging it wins 25 strata and wins 22, 24, and 25 strata for the individual seeds. Comparing the two complete primary losses gives the same direction, with differences 5.56 and 18.80 points and 28/28 and 24/28 stratum agreement, respectively. The 640-state results replace, rather than pool with, the smaller one-state-per-stratum diagnostic.

Objective-by-weighting factorial. All 2D training losses evaluate Δ_A and Z on the RMS-normalized residual $\tilde{r} = r/\mathrm{rms}(r)$; this leaves Equation 7 unchanged because Δ_A/Z is scale invariant, but it matters for the other cells. Let $\bar{Z}$ be the training-split mean of $Z(\tilde{r})$. The four objectives are

$$\mathcal{L}_{R,U} = \mathcal{L}_R, \quad \mathcal{L}_{R,Z^{-1}} = \frac{\bar{Z}}{Z}\mathcal{L}_R, \quad \mathcal{L}_{E,U} = 1 - \frac{\Delta_A(\tilde{r})}{\bar{Z}}, \quad \mathcal{L}_{E,Z^{-1}} = 1 - \frac{\Delta_A}{Z}, \tag{9}$$

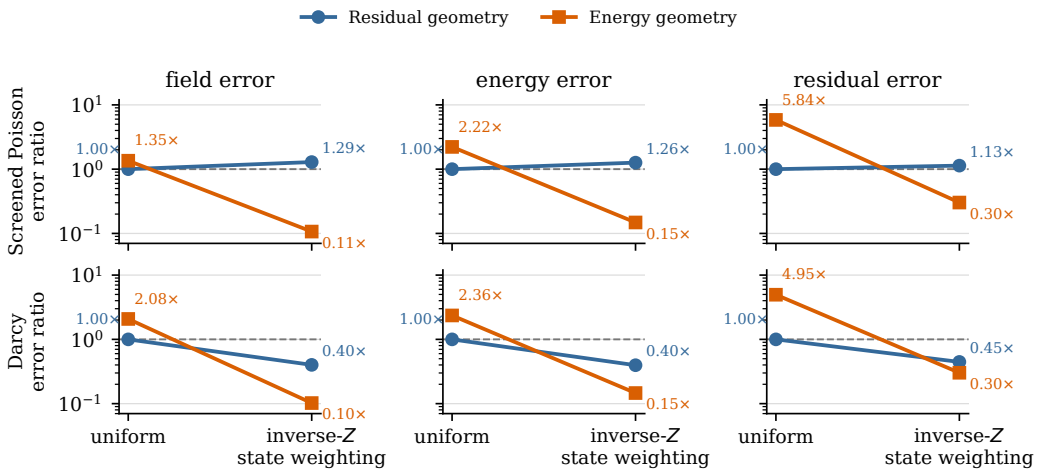

Figure 7: The complete E-DCDM loss is best on all three errors of both PDEs after eight operator products. Values are relative to Residual-DCDM for the validation-selected network of each cell. Uniform-energy training underperforms Residual-DCDM on both PDEs, while inverse-Z weighting of the residual loss helps on Darcy but not on screened Poisson. Exact values are in Tables 8 and 9.

and the last is E-DCDM. For any error metric Q, the multiplicative interaction is $I_Q = (Q_{E,Z^{-1}}/Q_{R,Z^{-1}})/(Q_{E,U}/Q_{R,U})$; values below one indicate a combination-specific gain beyond the two main effects. For the screened-Poisson row of Figure 7 we retrain all three seeds in the two new configurations, $\mathcal{L}_{R,Z^{-1}}$ and $\mathcal{L}_{E,U}$, on the exact shared matched recurrent-state training set. The constant $\bar{Z}$ only matches numerical scale; the weighting comes from $1/Z$, and $\bar{Z}$ changes neither relative state weights within a configuration nor the minimizer of its aggregate loss. Because $\Delta_A(\tilde{r}) = \Delta_A(r)/\mathrm{rms}(r)^2$, the uniform-energy cell weights states by $1/\|r\|_2^2$ (relative to the exact relative-energy loss, by $\|e\|_A^2/\|r\|_2^2$), and the inverse-Z cells use $Z(\tilde{r}) = Z(r)/\mathrm{rms}(r)^2$. The factorial does not test a raw residual magnitude: Residual-DCDM keeps its per-state $\|r\|_2^2$ denominator, so inverse-Z weighting is crossed only with this normalized residual loss. Validation selects seed 823 for uniform-energy training and seed 822 for inverse-Z-weighted residual training. On screened Poisson after eight operator products, neither isolated change improves any reported error; only their combination does. Thus the normalizer is not cosmetic, and its benefit there comes from its interaction with energy geometry rather than from a generally superior sampling weight. The field interaction computed within each seed is 0.070, 0.072, and 0.064 for seeds 821–823. The validation-selected cells come from different seeds; their interaction factors for field, energy, and residual are 0.061, 0.053, and 0.046, with instance-clustered 95% intervals $[0.053, 0.070]$, $[0.046, 0.060]$, and $[0.040, 0.052]$, which condition on those networks and do not cover seed variation. The conclusion is not a lucky additional-configuration checkpoint: for each isolated change, all three independently trained candidate seeds have worse field, energy, and residual endpoints than the validation-selected residual-uniform reference on screened Poisson. In the Darcy replication, all three inverse-Z-weighted residual seeds improve all three metrics for all 16 warm-start models, while all three uniform-energy seeds worsen all three metrics for every warm-start model. The selected combination remains best. Uniform-energy training does lower early field error, but its advantage disappears by seven operator products (Figure 8). Its Darcy field interaction is 0.100, 0.104, and 0.107 within seeds 811–813; the selected cells (residual-inverse-Z seed 813, uniform-energy seed 812) give 0.123 with instance-clustered 95% interval $[0.118, 0.128]$. Inverse-Z-weighted residual geometry improves field error by 59.9% for the selected checkpoint and 52.7–59.7% across the three seeds relative to each seed's own residual model; uniform-energy geometry worsens it by 107.8% for the selected checkpoint and 107.8–194.7% across seeds. The complete E-DCDM loss improves the selected field endpoint by 89.8%.

Same-network supervised reference. Replacing only the energy normalizer by the exact initial error energy $e^\top Ae = e^\top r$ gives the exact denominator unavailable to the proposed solution-free loss

Table 8: Only the complete E-DCDM loss improves every screened-Poisson endpoint. Entries are exact eight-product values.

Geometry	State weighting	E_2	E_A	E_R
Residual	uniform	$6.431e{-9}$	$2.708e{-9}$	$1.731e{-9}$
Residual	inverse-Z	$8.292e{-9}$	$3.421e{-9}$	$1.964e{-9}$
Energy	uniform	$8.701e{-9}$	$6.008e{-9}$	$1.010e{-8}$
Energy	inverse-Z (E-DCDM)	$\mathbf{6.875e{-10}}$	$\mathbf{4.010e{-10}}$	$\mathbf{5.234e{-10}}$

Table 9: Inverse-Z weighting helps residual geometry on Darcy, but the complete loss remains best. Entries are exact eight-product factorial values.

Geometry	State weighting	E_2	E_A	E_R
Residual	uniform	$6.432e{-6}$	$1.842e{-5}$	$1.403e{-4}$
Residual	inverse-Z	$2.579e{-6}$	$7.297e{-6}$	$6.255e{-5}$
Energy	uniform	$1.336e{-5}$	$4.339e{-5}$	$6.943e{-4}$
Energy	inverse-Z (E-DCDM)	$\mathbf{6.569e{-7}}$	$\mathbf{2.693e{-6}}$	$\mathbf{4.241e{-5}}$

and yields the solution-supervised post-step A-angle loss

$$1 - \frac{(p^\top Ae)^2}{(p^\top Ap)(e^\top Ae)}.$$

Because $Ae = r$, its direction-dependent quotient $(p^\top Ae)^2/(p^\top Ap)$ equals Δ_A, so it has the same per-state optimal direction as Equation 7, but its exact denominator changes cross-state weighting. This control reuses the earlier initial-state training set, with PCG labels for its 384 screened-Poisson training and validation states (maximum residual $9.995e{-13}$); seeds, epochs, batch size, optimizer, architecture, and validation rule match the initial-state arms. Across the same three screened-Poisson seeds, the selected model reaches $2.765e{-8}$ field error, versus $6.297e{-8}$ for the solution-free objective and $1.008e{-5}$ for residual training under the same 10^{-4} recipe. Under matched recurrent states the ranking reverses. A second control trains the exact normalizer on the matched states, with float64 conjugate-gradient labels (maximum true relative residual $1.3e{-13}$) and otherwise the Z_{32} recipe. Its field error exceeds that of Z_{32} on every seed: 1.81, 1.75, 2.38× on Darcy and 1.30, 1.61, 1.89× on screened Poisson (energy and residual errors agree), with selected-seed intervals $[2.31, 2.46]$ (Darcy seed 813) and $[1.22, 1.39]$ (screened seed 821). It still beats Residual-DCDM 4.0–5.6×. Because Z_{32} wins, its log gain over Residual-DCDM is 1.33–1.63 (Darcy) and 1.16–1.41 (screened) times that of the exact normalizer. A sealed prediction based on the initial-state result expected the exact normalizer to win and failed. A competing sealed prediction held on both counts: because the exact normalizer does not down-weight smooth-error states, it would do no better than Z_{32}, yet better than Z_{SD}, which up-weights them. Under the shared recipe, clipping confounds this comparison: the exact arm clips fewer batches (76–79% Darcy, 41–45% screened) than the audited Z_{32} seed (96%, 81%), and removing clipping from Z_{32} alone raises its screened error 1.11–1.76× (at 10^{-4}; Table 19). A follow-up therefore tuned the exact normalizer over the same nine-configuration screened-Poisson grid as the other losses (Table 10). Its selection (clipping at 10^{-3}, active on 1.1–1.5% of batches) remains 1.32, 1.41, and 1.80× worse than tuned Z_{32} (interval $[1.14, 1.52]$ at the validation-selected seed 821; energy 1.26–1.76×, residual 1.23–1.75×). All 15 unclipped configuration-matched ratios are at least 1.00 and 11 exceed 1.25; the runner-up and test-selected variants give 1.32–2.35× and 1.89–1.94×. A sealed prediction for this follow-up expected parity or a mixed result (geometric mean 0.9–1.3) and failed (1.49); on Darcy, the transferred settings give 1.48, 1.62, and 2.34× (Table 12), where parity or a mixed result was also predicted and failed. The result is consistent with the hardness-weighting hypothesis but does not identify which states' weights matter. Residual-DCDM, Z_{32}, the exact normalizer, and the step-overshoot control were tuned over the same screened-Poisson grid; the other normalizers and Darcy were not, and neither reference is a ceiling.

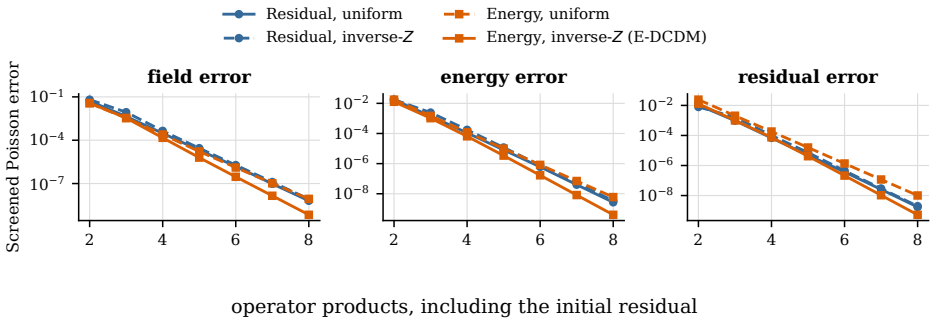

Figure 8: The complete E-DCDM loss separates from the other configurations after the first steps. All four screened-Poisson factorial configurations use the same matched recurrent-state training set; energy geometry with uniform state weights lowers early field error but loses that advantage by seven operator products, and the complete loss is lowest on all three metrics after eight.

Step-overshoot control. For any direction p, the residual loss $\mathcal{L}_{\mathrm{R}}(p) = \|r - \alpha(p)Ap\|_2^2/\|r\|_2^2$ of Equation 4 splits exactly as

$$\mathcal{L}_{\mathrm{R}}(p) = \mathcal{L}_{\mathrm{M}}(p) + (\alpha(p) - \alpha_{\mathrm{MR}})^2 \frac{\|Ap\|_2^2}{\|r\|_2^2}, \qquad \mathcal{L}_{\mathrm{M}}(p) = 1 - \frac{(r^\top Ap)^2}{\|Ap\|_2^2\|r\|_2^2},$$

with $\alpha_{\mathrm{MR}} = r^\top Ap/\|Ap\|_2^2$; both terms are invariant to rescaling r, so evaluating them on $\tilde{r}$ changes nothing. The step-overshoot control (M) trains $\mathcal{L}_{\mathrm{M}}$ on the same A-orthogonalized directions and matched states, so it removes exactly the overshoot penalty. Three consequences limit its reading: it is scored during training by a step it does not take at deployment (every model is rolled out with the shared energy step); it is consistent only per direction, since a residual-optimal method over the whole span would need $A^\top A$-orthogonal history (conjugate residual or GCR) trained on its own rollouts; and deployment with the minimal-residual step is not run, because it breaks the orthogonality of the residual to the search space and moves the states off the training views. The predeclared bands for the closed fraction f were $f \geq 0.8$ (the penalty explains the gap), $f \leq 0.2$ (it does not), and $0.2 < f < 0.8$ (both contribute), read on all three seeds under the shared recipe and with each loss tuned, and requiring a one-step check before the "does not explain" reading. Under the shared recipe $f = 0.03, 0.11, 0.18$ (interval $[0.14, 0.21]$ at seed 823); the one-step check (Table 11) shows lower realized energy decrease than E-DCDM but a deployed-step overshoot 1.24–1.69 times Residual-DCDM's, so that reading is not confirmed; at its own step the control leaves 8–22% less residual than Residual-DCDM leaves at the energy step, and within about 12% of Residual-DCDM at its own minimal-residual step. Tuned, $f = 0.12, 0.52, 0.78$ (interval $[0.71, 0.85]$ at seed 823), spanning bands. The control selected 10^{-3} without clipping and beats Residual-DCDM on 23 of the 27 configuration-seed cells, but never matches E-DCDM ($M/E \geq 1.44$). A sealed prediction expected both to contribute, with f of 0.3–0.6 under the shared recipe and lower when tuned; the shared-recipe values were lower and the tuned ones higher. On Darcy at the shared recipe the control closes 0.34, 0.45, and 0.42 of the gap ($M/E = 3.5$–4.0; sealed prediction 0.2–0.6), with an overshoot 3.4–3.9 times Residual-DCDM's. Under the shared recipe the three screened-Poisson arms also clip differently (control 25–34%, Residual-DCDM 51%, E-DCDM 81% of batches for the audited seeds).

Table 10: Screened-Poisson grid for the tuned exact normalizer (X) and the step-overshoot control (M), seeds 821–823. Field errors are geometric means over seeds; X/Z_{32} and M/E are per-seed ratios (min–max) to E-DCDM (Z_{32}) at the same configuration; f is the closed fraction of the log field-error gap for seeds 821/822/823; validation is each arm's mean best validation objective; $\star$ marks each selection. The clipping-on, 10^{-4} exact row reuses the earlier exact control.

Clip	Rate	X field	X/Z_{32}	X valid.	M field	M/E	f	M valid.
on	10^{-4}	$1.37e{-9}$	1.30–1.89	.00320	$5.10e{-9}$	5.49–6.17	.03/.11/.18	.00349
on	3×10^{-4}	$8.66e{-10}$	1.54–1.61	.00229	$1.76e{-9}$	2.11–4.09	.13/.19/.43	.00222
on	10^{-3}	$1.25e{-9}$	1.62–2.35	.00221*	$1.73e{-9}$	1.86–4.33	.04/.48/.50	.00168
on	3×10^{-3}	$1.37e{-9}$	0.94–1.55	.00241	$3.46e{-9}$	2.19–4.98	−.35/−.37/.28	.00194
off	10^{-4}	$1.67e{-9}$	1.04–1.64	.00339	$5.43e{-9}$	2.94–7.27	.29/.32/.06	.00367
off	3×10^{-4}	$9.34e{-10}$	1.99–2.11	.00234	$2.19e{-9}$	3.58–6.27	.19/.03/.26	.00224
off	10^{-3}	$1.29e{-9}$	1.32–1.68	.00227	$1.76e{-9}$	1.44–3.70	.12/.52/.78	.00162*
off	3×10^{-3}	$1.60e{-9}$	1.00–1.78	.00245	$2.66e{-9}$	1.67–2.41	.60/.11/.30	.00191
off	10^{-2}	$2.23e{-9}$	1.11–1.50	.00266	$1.90e{-8}$	2.81–57.7	−.14/.33/−.58	.00377

Table 11: One-step diagnostic on the shared-view validation states (not rollouts of each model), seed ranges. "Energy left" is $1 - \Delta_A/\|e\|_A^2$ after the energy step; "overshoot" is the penalty $(\alpha(p) - \alpha_{\mathrm{MR}})^2\|Ap\|_2^2/\|r\|_2^2$ (Appendix A.5); residuals are relative squared residual norms after the energy step and after the minimal-residual step. Shared recipe: clipping, 10^{-4}; tuned: each arm's selection.

Setting	Model	Energy left	Overshoot	Residual (energy step)	Residual (MR step)
Shared	Residual-DCDM	.0046–.0050	.00039–.00056	.0039–.0046	.0035–.0040
Shared	Control (M)	.0043–.0051	.00058–.00075	.0039–.0044	.0033–.0036
Shared	E-DCDM	.0034–.0037	.00081–.00085	.0065–.0068	.0056–.0060
Tuned	Residual-DCDM	.0030–.0035	.00023–.00026	.0019–.0020	.0017–.0018
Tuned	Control (M)	.0031–.0032	.00030–.00035	.0019–.0020	.0016–.0017
Tuned	E-DCDM	.0023	.00030–.00045	.0033–.0037	.0029–.0032
Tuned	Exact normalizer	.0022–.0023	.00029–.00037	.0030–.0032	.0027–.0028

Table 12: Darcy with the screened-Poisson selections transferred (not tuned on Darcy), seeds 811–813. Ratios are same-seed eight-product field errors; shared-recipe values in parentheses. The step-overshoot control uses the shared recipe.

Quantity	811	812	813
E-DCDM/Residual-DCDM	.129 (.120)	.118 (.102)	.112 (.104)
Exact/Z_{32}	1.48 (1.81)	1.62 (1.75)	2.34 (2.38)
Exact clipped batches	2.6%	2.5%	2.8%
Control (M): f	.34	.45	.42
Control (M): M/E	4.02	3.49	3.72
Control (M): M/Residual	.48	.36	.39

A.6 NORMALIZER SENSITIVITY

Coarse-normalizer sensitivity. We replace Z_{32} by the energy captured in the first 8 or 64 nested sine modes, or by $Z_{\mathrm{SD}} = (r^\top r)^2/(r^\top Ar)$, the exact energy decrease of a steepest-descent step. The latter is solution-free but contains no coarse spectral model. Every alternative is trained from the paired initialization on the same recurrent-state training set with all three seeds. The raw objective retains its native inverse weight; the mean-matched variant rescales that weight to match the training-split mean of $1/Z_{32}$.

Figure 9 shows that the difference emerges late in recurrence: Z_8 is slightly worse after the first few Darcy operator products but overtakes Z_{32} after four or five operator products, while Z_{64} crosses in the opposite direction and Z_{SD} compounds sharply. A one-seed validation-only interpolation at 16 modes improves the validation mean of the directional term Δ_A/Z over Z_{32} by 29.7% on Darcy and 31.1% on screened Poisson before mean matching, but is 3.0% and 6.4% worse after mean matching. Held-out test roles were not opened for this diagnostic, so it describes optimization and scale, not endpoint performance. On Darcy, Z_8 improves field, energy, and residual endpoints and wins at least 79.7% of instances and all 16 warm-start models for every seed and scale. Screened-Poisson eight-product field point estimates also favor Z_8 for every seed, but the selected intervals overlap equality.

No Z_{32} value activates the 10^{-20} training floor or the stricter 10^{-18} training-set generation rejection rule. The largest one percent of nominal weights also remains below the predeclared 25% concentration gate, as does the largest one percent of realized per-example gradient norms at all three audited checkpoints. Clipping nevertheless affects most training batches of both losses under the shared recipe (Table 14). The tuned Z_{32} and residual selections disable clipping; at rate 10^{-3} with clipping, it affects at most 9% of E-DCDM batches, and E-DCDM wins every seed in every cell (Table 19). The tuned exact normalizer selects clipping at 10^{-3}, where it acts on 1.1–1.5% of batches, and it is never better than Z_{32} at any unclipped matched configuration (1.00–$2.11\times$; 11 of 15 above 1.25), so clipping does not explain the exact-normalizer gap.

Thus Z_{SD} is worse than same-seed Residual-DCDM on every Darcy seed under both scalings; on screened Poisson the same-seed comparison is mixed under both scalings, and Z_{SD} remains 5.3–7.5 times worse than same-seed Z_{32}. These results establish that the tested coarse spectral normalizer is not reproduced by the tested solution-free energy scale. They also show that 32 modes are not insensitive and are not optimal for the frozen field endpoint after eight operator products among the tested sizes. Because Z_8 was identified in this held-out diagnostic, we do not promote it to the primary method without a new independent test.

Basis-family control. We additionally froze one equally sized alternative before constructing its normalizer: tensor products of four and eight piecewise-linear hats whose centers span the first through last interior grid points. Table 16 compares this local 32-vector dictionary with the same-seed global sine-32 reference. None of the four PDE-by-scale cells meets the predeclared practical-equivalence rule. Three cells change direction across seeds; only the mean-matched screened-Poisson cell consistently favors local hats. The largest one percent of local inverse weights carries 11.5% and 11.3% of Darcy and screened-Poisson weight, effective sample fractions are 25.7% and 22.2%, and no state reaches the denominator floor. Conversely, 79.0–98.6% of training batches clip across the four cells and three seeds. Thus this experiment does not isolate basis geometry from optimization under the fixed clipping rule, and the selected-seed case intervals do not replace the visible seed spread.

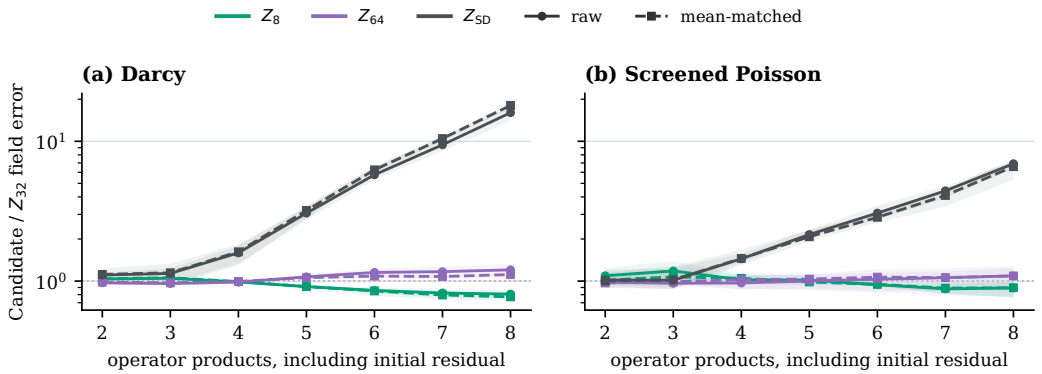

Figure 9: The tested low-mode normalizers outperform Z_{SD}, while spectral resolution controls late recurrent accuracy. Curves show the mean candidate-to-Z_{32} field-error ratio across three paired seeds; bands span those seeds. Ratios below one favor the candidate. Raw and mean-matched objectives give the same qualitative conclusion.

Table 13: No tested denominator is practically equivalent to Z_{32}. Entries are eight-product field-error ratios to the same-seed Z_{32} model; parentheses give the paired-instance 95% interval for the checkpoint selected using candidate validation loss (Darcy: seed 812 for Z_8 and Z_{64}, 813 for Z_{SD}; screened: 823 except Z_8 mean-matched 821 and Z_{SD} raw 821). Practical equivalence, fixed before evaluation, requires all three ratios and the interval to lie in $[0.9, 1.1]$.

Normalizer	Scale	Darcy seeds 811, 812, 813	Screened seeds 821, 822, 823
Z_8	raw	.729, .857, .835 ([.841, .873])	.996, .765, .920 ([.833, 1.020])
	mean-matched	.753, .754, .807 ([.738, .771])	.937, .763, .989 ([.837, 1.059])
Z_{64}	raw	1.039, 1.248, 1.323 ([1.231, 1.265])	1.169, .860, 1.240 ([1.174, 1.308])
	mean-matched	1.112, 1.110, 1.121 ([1.096, 1.123])	1.220, .772, 1.274 ([1.208, 1.344])
Z_{SD}	raw	16.231, 17.852, 13.969 ([13.456, 14.493])	7.201, 6.410, 7.127 ([6.423, 8.115])
	mean-matched	17.125, 18.319, 18.698 ([18.018, 19.401])	5.308, 7.497, 6.955 ([6.494, 7.461])

Table 14: Inverse-Z_{32} weights are heterogeneous but not top-1%-dominated; clipping is frequent. ESS/n is effective sample size divided by state count. Gradient shares are maxima over initial, epoch-500, and final validation checkpoints ("Top 1% grad."). "Hypoth. scaled" applies the configured batch clipping hypothetically to validation batches. Clipped-batch fractions and median multipliers are observed during training: E-DCDM (E) and Residual-DCDM (R) from the seed-811 and seed-821 training replays, which are tensor-identical to the reported checkpoints; the exact normalizer (X) over its three seeds.

Problem	Top 1% weight	ESS/n	Top 1% grad.	Hypoth. scaled	Clipped (E/R/X)	Multiplier (E/R/X)
Darcy	13.5%	20.5%	19.6%	12.3%	96/82/76–79%	.24/.57/.40–.42
Screened	12.3%	19.6%	12.5%	9.7%	81/51/41–45%	.28/.53/.54–.58

Table 15: The generic Z_{SD} scale can lose even to Residual-DCDM. Entries are field errors after eight operator products at each Z_{SD} arm's validation-selected seed (Darcy 813; screened 821 raw, 823 mean-matched), so the Residual-DCDM and Z_{32} columns are same-seed references and differ from Table 2; lower is better.

Problem	Scale	Residual-DCDM	Z_{SD}	Z_{32}
Darcy	raw	$6.407e{-6}$	$9.348e{-6}$	$6.692e{-7}$
Darcy	mean-matched	$6.407e{-6}$	$1.251e{-5}$	$6.692e{-7}$
Screened	raw	$6.437e{-9}$	$7.067e{-9}$	$9.814e{-10}$
Screened	mean-matched	$6.148e{-9}$	$4.782e{-9}$	$6.875e{-10}$

Table 16: The local-hat normalizer is not robustly equivalent to sine-32. Entries are eight-product field-error ratios for all paired seeds; parentheses give the validation-selected paired-instance 95% interval. Ratios below one favor local hats.

Problem	Scale	Local-hat-32 / sine-32
Darcy	raw	$1.010, .859, .824$ ($[.844, .875]$)
Darcy	mean-matched	$.810, .880, 1.050$ ($[1.024, 1.076]$)
Screened	raw	$.717, 1.462, .942$ ($[.882, 1.007]$)
Screened	mean-matched	$.695, .664, .788$ ($[.739, .840]$)

A.7 TRANSFER AND FAILURE BOUNDARIES

Unless stated otherwise, these stress tests use the initial-state checkpoints; the matched-state resolution transfer and the mixed objective schedule use matched recurrent-state checkpoints.

Resolution-transfer values. Table 17 reports the exact ratios summarized by Figure 5. The transferred fields are 256 masks per grid from `darcy_test_64.pt`, not the 64^2 evaluation fields (Appendix A.2). Every transferred condition favors E-DCDM in field and energy error but reverses in residual error. All 16 warm-start model means and four balanced warm-start cell means favor field error in every table row; the final column gives instance-level field wins.

Table 17: Exact eight-product resolution results. Entries are E-DCDM error divided by Residual-DCDM error; values below one favor E-DCDM. The 64^2 point in Figure 5a is the native-grid initial-state result of Table 2.

Checkpoint	Evaluation	Field	Energy	Residual	Field wins
Initial state	128^2 zero-shot	.318	.367	1.096	$242/256$
Initial state	256^2 zero-shot	.708	.750	1.576	$194/256$
Matched recurrent	128^2 zero-shot	.421	.465	1.245	$255/256$
Matched recurrent	256^2 zero-shot	.819	.831	2.257	$169/256$
Matched recurrent	128^2 native	.055	.096	.800	$256/256$

The selected matched-recurrent transfer was post hoc, with analysis frozen before scoring; its protocol would have withheld a failure on both grids, and both grids passed and are released. For the initial-state checkpoint at 256^2, squared spectral-coefficient sums are 0.00452 versus 0.0168 for low-frequency field-error power and $1.55e5$ versus $4.17e4$ for high-frequency residual power (E-DCDM versus Residual-DCDM). Absolute field errors increase by orders of magnitude under transfer. In the native-128^2 condition, absolute cross-condition comparisons are not monotone and both the training set and state coverage change, so the result does not identify a retraining effect.

External coefficient transfer. We fix the validation-selected initial-state checkpoints (residual seed 811 and energy seed 812) and evaluate 256 public CNO Darcy coefficients (Raonic et al., 2023), balanced across its in- and out-of-distribution partitions and crossed with the same 16 warm-start models. This population had previously been used in exploratory work, but not for training, selection, or adaptation, so this is external-covariate transfer rather than a prospective test. After eight operator products, energy training lowers field, energy, and residual errors by 92.25%, 85.57%, and 76.51%; field error improves for every instance.

Coefficient-contrast shifts. When Darcy contrast doubles from four to eight, the unchanged E-DCDM checkpoint reaches field error 3.816×10^{-4} versus 1.354×10^{-3} for Residual-DCDM, a 71.82% reduction with paired interval $[9.441, 10.019] \times 10^{-4}$. It lowers energy and residual error by 62.56% and 36.65%; all warm-start model and cell means improve, and field error improves on 99.7% of instances. This is a conditioning shift, not a new PDE or grid.

For the exploratory extreme-contrast boundary, we reuse the 1,000 public Darcy masks but change coefficient values from $(3, 12)$ to $(3, 300)$. Batched PCG targets have maximum true relative residual $9.29e{-}12$. Neither fixed learned recurrence is usable: field errors after eight operator products are 2.909 for Residual-DCDM and 11.457 for E-DCDM, and all 16 model means favor the residual-trained model. This negative result was generated after the contrast-eight outcome and is not a prospective test.

Mixed-objective schedule. For a seven-correction rollout, EmRn denotes m E-DCDM corrections followed by n Residual-DCDM corrections, with $m + n = 7$; E7 and R7 are the two unmixed endpoints.

E6R1 is the only tested switch depth passing at all three grids. This selection is descriptive and post hoc; a future study must choose a schedule using development data before claiming adaptivity or optimality.

Table 18: Only E6R1 satisfies the reporting rule at every tested grid. Field gain retained is relative to all-energy E7 versus all-residual R7; residual improvement is relative to E7. The post hoc reporting heuristic requires both residual improvement and at least 90% field-gain retention.

Grid	Schedule	Field error	Gain retained	Residual improvement	Pass
64^2	E6R1	$1.194e{-6}$	90.7%	4.9%	yes
	E5R2	$1.608e{-6}$	83.5%	-12.8%	no
128^2	E6R1	$8.830e{-4}$	97.6%	28.2%	yes
	E5R2	$9.896e{-4}$	88.5%	21.7%	no
256^2	E6R1	$2.059e{-2}$	101.5%	32.3%	yes
	E5R2	$2.107e{-2}$	91.1%	33.5%	yes

A.8 Robustness and classical baselines

Learning rate. On screened Poisson in the initial-state condition, with both objectives starting from identical seed-823 weights and changing only Adam's rate, matched rates $\{3{\times}10^{-5}, 10^{-4}, 3{\times}10^{-4}\}$ give field-error ratios of E-DCDM to Residual-DCDM of $1/30.7$, $1/160$, and $1/218$; both objectives improve at the largest rate. Under matched recurrent states the trend reverses: with clipping, the geometric-mean ratio rises from 0.136 at 10^{-4} to 0.217 at 3×10^{-4} and 0.242 at 10^{-3}. Table 19 gives the full screened-Poisson grid, extended to 10^{-2} without clipping (had clipping been on at 10^{-2}, it would have acted on 0.03–0.4% of Residual-DCDM and 1.7–1.9% of E-DCDM batches). An initial six-configuration grid, whose selections sat at its largest rate, was extended to these nine configurations; each loss again selected 10^{-3} without clipping, now an interior point. Per seed, the nine fixed configurations give ratios 0.11–0.43, and 0.03 on one seed at 10^{-2}, where Residual-DCDM trains poorly. A sealed prediction that Residual-DCDM would move to 3×10^{-3} failed, as did the predicted tuned ratio of 0.25–0.40 (observed geometric mean 0.233) and the predicted sensitivity to selection. For both losses the runner-up (clipping, 10^{-3}) trails by about 1.9 across-seed standard errors, and no other configuration lies within one. Substituting E-DCDM's runner-up gives ratios 0.14–0.26 and Residual-DCDM's gives 0.26–0.38; the test-selected pair (post hoc) gives 0.15–0.22, and Residual-DCDM's test-best configuration alone gives 0.25–0.49. The paired interval for the tuned ratio at the validation-selected seed 823 is $[0.168, 0.206]$. Transferring the selections to Darcy gives 0.129, 0.118, and 0.112 on seeds 811–813 (shared recipe 0.120, 0.102, 0.104; Table 12), within the sealed prediction of 0.10–0.30; Darcy was not tuned.

Table 19: Extended screened-Poisson optimizer grid under matched recurrent states (seeds 821–823). Field error after eight products is the geometric mean over seeds; "Ratio" is E-DCDM/Residual-DCDM per seed (min–max). Validation is the mean best validation objective used for selection (lower is better; the two losses use different objectives). The clipping-on, 10^{-4} row reuses the primary checkpoints; $\star$ marks each loss's selection.

Clip	Rate	E-DCDM field	Residual field	Ratio	E-DCDM valid.	Residual valid.
on	10^{-4}	$8.65e{-10}$	$6.34e{-9}$	.112–.152	-1.54346	.00422
on	3×10^{-4}	$5.48e{-10}$	$2.52e{-9}$	.176–.270	-1.54560	.00247
on	10^{-3}	$6.66e{-10}$	$2.75e{-9}$	.217–.293	-1.54568	.00204
on	3×10^{-3}	$1.07e{-9}$	$3.00e{-9}$	.311–.434	-1.54520	.00240
off	10^{-4}	$1.23e{-9}$	$8.06e{-9}$	.122–.205	-1.54273	.00457
off	3×10^{-4}	$4.52e{-10}$	$2.96e{-9}$	.133–.177	-1.54547	.00254
off	10^{-3}	$8.34e{-10}$	$3.59e{-9}$	.186–.301	$-1.54575^\star$	$.00196^\star$
off	3×10^{-3}	$1.26e{-9}$	$4.04e{-9}$	.274–.370	-1.54508	.00228
off	10^{-2}	$1.73e{-9}$	$1.47e{-8}$	.029–.265	-1.54403	.00354

Limited-memory recurrence. The released DCDM recurrence keeps only the two most recent direction/product pairs. Re-evaluating the same fixed initial-state checkpoints with this two-memory rule raises both methods' errors, but energy training still lowers eight-product field error by 84.1% on Darcy ($1.552e{-5}$ versus $9.744e{-5}$) and 99.2% on screened Poisson ($2.237e{-7}$ versus $2.898e{-5}$). After 12 operator products it remains 84.9% and 99.95% lower, respectively. Full memory strengthens the absolute solver, but it is not the source of the objective gap.

Fixed spectral outputs. Although 95–96% of late squared field error lies in 32 sine modes, an energy-trained model with fixed spectral outputs is 2.4% (Darcy) and 31.1% (screened Poisson) worse than full-memory Residual-DCDM after seven corrections. All arms of this control train on the same full-memory states after six Residual-DCDM corrections (three seeds, 500 epochs, batch size 64) with the same placement and line update; the reported spectral model, like the dense model, is trained through the same cached-span projection. It has 99,493 active parameters, 0.25% below the dense 99,737 (its 23,941 unused routing parameters receive no gradient; an earlier width-48 version had 44,473); output parameterization and preprocessing necessarily still differ, so matching capacity and projection does not isolate every architectural choice. Every energy-trained dense seed instead improves on Residual-DCDM for every instance, by 65.4–66.1% on Darcy and 77.6–79.5% on screened Poisson (also in every screened stratum). Of the three residual-trained dense Darcy seeds, one is indistinguishable from Residual-DCDM and the other two lower its field error by 36.5% and 9.1%. In the tested architectures, dense coefficient-adaptive directions beat hard-coded spectral ones, and the energy-trained dense model is both better and more consistent across seeds.

Arithmetic cost of a network call. Table 20 compares one direction-network call with one variable-coefficient stencil product. Network floating-point operations are counted with PyTorch's FLOP counter (two per multiply-add); stencil counts are analytic (9 per unknown for the 5-point and 13 for the 7-point operator). In arithmetic, one call equals about 1.5×10^4 operator products in 2D and 5.3×10^3 in 3D, so a six-call learned solve to 10^{-3} costs roughly 9×10^4 operator products of arithmetic, against a few hundred matvec-equivalents for CG (180 products plus vector updates). Measured GPU times are closer because small stencil products are dominated by launch overhead.

Table 20: One network call is far more arithmetic than one operator product. Times are medians of 200 synchronized float32 calls on one RTX 8000.

Grid	Network GFLOP	Ratio to Av	Network ms (batch 1 / 32)	Av ms (batch 1 / 32)
62^2 (64^2 grid)	0.52	1.5×10^4	1.34 / 3.37	0.15 / 0.15
126^2 (128^2 grid)	2.15	1.5×10^4	1.34 / 14.4	0.15 / 0.16
32^3	2.27	5.3×10^3	1.52 / 28.8	0.22 / 0.38
64^3	18.2	5.3×10^3	7.57 / 242	0.22 / 3.12

Full-population work to accuracy. After fixing the analysis and promotion rule, we extended all paired matched-state checkpoints from eight to 32 total operator products on the complete 2D evaluation populations. Products 9–32 extrapolate beyond the training depths. All cases reach every prespecified field- and energy-error tolerance by product 32 for both learned objectives. Table 21 reports the paired difference in mean censored hitting time; negative values favor E-DCDM. Darcy passes its problem-level rule: every threshold favors E-DCDM, and the required gain of at least one product for every seed occurs at 10^{-5} and 10^{-6} for both metrics. Screened Poisson also favors E-DCDM at every threshold, with identity-bootstrap intervals excluding zero, but its gains are only 0.02–0.57 products, so no threshold reaches the predeclared one-product-per-seed relevance requirement.

The cross-problem rule therefore fails. Its exact no-late-reversal clause also fails for screened seed 821 at product 32: E-DCDM is higher by 1.35×10^{-16} in field error and 6.88×10^{-18} in energy error, when both methods are already at approximately 10^{-12} error. We retain the failure rather than waive it as numerical tolerance. We consequently treat this as mixed post hoc evidence, not confirmation of a cross-problem work advantage. The intervals resample shared identities and condition on the six trained checkpoint pairs. The released receipt additionally gives reached fractions, censored means and quantiles, endpoints, residual-stopping field errors, and the CG and Jacobi-PCG references.

Classical baselines and time to tolerance. Table 22 compares operator products and measured time to residual tolerance on one fixed 32-case Darcy batch with the validation-selected matched recurrent-state checkpoints. Each time is the median of 600 repeats; each repeat times all methods once in random order, synchronizing the device before each call; the block medians are released. An earlier run that timed each method in one contiguous block gave 1.18 versus 0.73 ms per case (E-DCDM versus Residual-DCDM) at 10^{-3} and 0.48 versus 0.71 at 10^{-2}, and its CG, Jacobi-PCG, and single-operator-product times were 1.7–2.1 times the interleaved values; its 10^{-3} gap does not reproduce,

Table 21: The 32-product audit passes its Darcy rule but not its cross-problem promotion rule. Entries are E-DCDM minus Residual-DCDM mean total products to tolerance for each paired seed; negative values favor E-DCDM. The last column is the 95% shared-identity bootstrap interval for the seed-averaged difference. All cases reach every listed tolerance by product 32.

Metric	Tolerance	Seed 1	Seed 2	Seed 3	95% interval
Darcy (seeds 811–813)					
Field	10^{-2}	-0.53	-0.55	-0.37	$[-0.50, -0.46]$
Field	10^{-3}	-0.59	-0.63	-0.59	$[-0.62, -0.59]$
Field	10^{-4}	-0.89	-0.90	-0.91	$[-0.91, -0.89]$
Field	10^{-5}	-1.08	-1.11	-1.10	$[-1.11, -1.09]$
Field	10^{-6}	-1.26	-1.37	-1.38	$[-1.35, -1.32]$
Energy	10^{-2}	-0.64	-0.69	-0.51	$[-0.63, -0.60]$
Energy	10^{-3}	-0.83	-0.85	-0.74	$[-0.82, -0.79]$
Energy	10^{-4}	-0.85	-0.94	-0.91	$[-0.92, -0.89]$
Energy	10^{-5}	-1.02	-1.09	-1.11	$[-1.09, -1.06]$
Energy	10^{-6}	-1.12	-1.16	-1.23	$[-1.18, -1.15]$
Screened Poisson (seeds 821–823)					
Field	10^{-2}	-0.28	-0.27	-0.25	$[-0.31, -0.22]$
Field	10^{-3}	-0.23	-0.29	-0.29	$[-0.31, -0.23]$
Field	10^{-4}	-0.21	-0.27	-0.26	$[-0.28, -0.21]$
Field	10^{-5}	-0.26	-0.24	-0.22	$[-0.28, -0.20]$
Field	10^{-6}	-0.38	-0.36	-0.40	$[-0.43, -0.33]$
Energy	10^{-2}	-0.02	-0.15	-0.04	$[-0.08, -0.05]$
Energy	10^{-3}	-0.03	-0.08	-0.03	$[-0.06, -0.03]$
Energy	10^{-4}	-0.12	-0.11	-0.11	$[-0.14, -0.09]$
Energy	10^{-5}	-0.30	-0.29	-0.33	$[-0.37, -0.25]$
Energy	10^{-6}	-0.52	-0.48	-0.57	$[-0.57, -0.47]$

so all entries come from the interleaved run. These single-GPU measurements are not cross-platform claims and cover one batch and residual tolerances only; a multi-batch error-versus-time study is future work.

Table 22: The learned solvers need far fewer operator products than CG and Jacobi-PCG to reach residual tolerance. Products exclude the common product that forms the initial residual; each count is the smallest number at which at least 16 of the 32 Darcy cases reach the tolerance. Times are median milliseconds per case over 600 randomly interleaved repeats on one RTX 8000; they include per-step network inference, vector work, and Jacobi diagonal construction, and exclude checkpoint loading and the common warm-start inference. All 32 cases reach each tolerance for every method; the learned solvers need 3–7 products per case. Paired bootstrap 95% intervals for the E-DCDM/Residual-DCDM time ratio are $[0.992, 1.002]$ at 10^{-3} and $[0.793, 0.801]$ at 10^{-2}. AMG-PCG results follow the table.

	10^{-2}		10^{-3}	
Method	Products	ms/case	Products	ms/case
E-DCDM	**4**	**0.486**	**6**	**0.735**
Residual-DCDM	5	0.610	**6**	**0.737**
Jacobi-PCG	94	0.993	125	1.319
CG	140	1.442	180	1.850

At the same eight-product count, seven-step vanilla CG and Jacobi-PCG give respectively 7.982×10^{-2} and 7.543×10^{-2} on Darcy, and both give 2.968×10^{-1} on screened Poisson. Starting instead after six Residual-DCDM corrections, raw residual, projected residual, raw Jacobi, and projected Jacobi tails reach $(2.397, 2.306, 2.386, 2.276) \times 10^{-4}$ on Darcy and $(5.133, 4.943, 5.133, 4.943) \times 10^{-5}$ on screened Poisson. These controls use exact line steps and the identical warm starts, prefixes, and operator-product count.

Known-stencil exact correction in the first 32 sine modes reaches 4.630×10^{-5} on Darcy, between spectral and dense energy tails. AMG-PCG reaches 5.313×10^{-5} after three V-cycles and 9.157×10^{-10} after seven, with hierarchy setup charged separately. For the coefficient-specific timing study, assembly and hierarchy setup take median 1.30 and 22.51 ms, and 16 one-cycle solves together take 23.58 ms on one CPU thread. These controls delimit the learned-direction results under a product-

limited matrix-free interface; a reusable hierarchy or unrestricted iteration should be preferred when available.