AMR-VNO: LEARNING ON BLOCK-STRUCTURED ADAPTIVE MESH HIERARCHIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive mesh refinement (AMR) solvers store solutions on nested levels of cells, yet most grid-based neural operators first rasterize these fields onto a uniform grid. We introduce AMR-VNO (AMR V-cycle Neural Operator), whose computational graph is the solver's block-AMR hierarchy, extended where needed by virtual coarse levels built from the inputs. It exchanges direction-aware messages within each level, moves information between levels through learned restriction and prolongation shared across levels, and predicts directly on active leaf cells. On the evaluated held-out 2D Darcy and advection-diffusion sets, AMR-VNO has 14 to 20% lower mean error against the solver's discrete targets than a tuned multigrid neural operator, with about 34 times fewer learned parameters, and lower error than graph and sparse-convolution models given the same hierarchy. It also improves over same-hierarchy message passing on in-distribution 3D conduction. On public Quokka turbulence, AMR-VNO and hierarchy message passing both outperform dense baselines but are not statistically distinguishable. In Darcy validation ablations, direction-aware aggregation and a second traversal each reduce error. Accuracy degrades on deeper hierarchies and changed refinement policies, so sharing weights across levels does not by itself give transfer across hierarchy structure.

1 INTRODUCTION

Neural operators learn maps between functions and can amortize repeated PDE solves (Lu et al., 2021; Li et al., 2020a; Kovachki et al., 2023). Most common implementations assume a fixed lattice: Fourier neural operators (FNOs), for example, use FFT-based layers on uniform arrays (Li et al., 2020a). Many scientific solvers instead produce nested patch hierarchies.

Adaptive mesh refinement places fine cells only where a refinement criterion requests them (Berger & Oliger, 1984; Berger & Colella, 1989). Block-structured frameworks such as AMReX store levels, rectangular patches, refinement ratios, and active leaf cells (Zhang et al., 2019). Rasterizing such a field at the finest resolution spends computation on cells the solver chose not to refine and discards its parent-child structure, while a coarse raster discards the fine values that motivated refinement. On our HeatTransport2D benchmark, active leaves average about 23% of the finest 512^2 grid cells.

An AMR hierarchy already has the structure a multigrid cycle needs: nested levels, parent-child links, and fine cells only where the solver placed them. AMR-VNO uses this structure directly as its computational graph. We ask whether a model sharing weights across levels can consume the solver hierarchy and predict directly on its active leaves. Our contributions are:

1. *Method.* AMR-VNO computes directly on the solver's block-AMR hierarchy, extended by virtual coarse levels on the 2D and Conduction3D benchmarks, with a direction-aware V-cycle. Its weights are shared across levels and cycles, so its learned parameter count does not grow with depth, and it predicts only on active leaves.
2. *Controlled findings.* We compare against dense, multigrid, sparse-convolution, and same-hierarchy graph models on held-out sets. Among the evaluated controls, access to the same hierarchy did not reproduce AMR-VNO's gains on the 2D benchmarks or in-distribution 3D conduction. On Quokka, AMR-VNO and hierarchy message passing are not statistically distinguishable, and accuracy degrades on deeper hierarchies.

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.

3. *Benchmarks.* We release four AMR benchmarks, including a trajectory-disjoint benchmark from public Quokka turbulence with solver-native dynamic regridding.

Our results suggest that the solver's hierarchy is a useful computational representation; making such models robust to how the hierarchy is built remains open.

2 Related Work

Uniform-grid operators. FNO learns global spectral filters on regular grids (Li et al., 2020a), while CNO uses alias-controlled convolutional representations (Raonic et al., 2023). Multi-resolution operators such as U-NO (Rahman et al., 2023) and MgNO (He et al., 2024) build internal coarse-to-fine features from a uniform input. Their hierarchy is chosen by the network, ours by the PDE solver.

Irregular representations. NUNO partitions nonuniform samples into regular subdomains (Liu et al., 2023), while the Multipole Graph Neural Operator constructs a multilevel inducing-point graph for long-range interactions (Li et al., 2020b). Geo-FNO maps irregular domains to a latent regular representation (Li et al., 2023a), and GINO connects geometric inputs and outputs through graph and Fourier operators (Li et al., 2023b). Recent point-cloud, multiscale, and topological operators broaden the representations available beyond fixed lattices (Zeng et al., 2025; Han et al., 2026; Pei et al., 2026; Bastian et al., 2026). None of these operates on a solver-emitted block-AMR hierarchy, which AMR-VNO uses as both its computational and its output representation.

Mesh adaptation. MeshGraphNets predicts dynamics on unstructured meshes and can adapt its discretization (Pfaff et al., 2021). LAMP jointly learns a dynamics surrogate and a policy for refinement and coarsening (Wu et al., 2023), ASMR learns a refinement policy for finite-element meshes (Freymuth et al., 2023), and AMBER predicts sizing fields and iteratively remeshes toward expert meshes (Freymuth et al., 2025). These methods change the mesh; AMR-VNO takes it as given.

Multiscale and adaptive mesh learning. BSMS-GNN and MultiScale MeshGraphNets construct multilevel graphs from input meshes (Cao et al., 2023; Fortunato et al., 2022), and Graph U-Nets learn pooling scores (Gao & Ji, 2019); we compare against task-adapted graph U-Net and hierarchy-MPNN controls and a validation-only BSMS-GNN adaptation. Most directly, Perera & Agrawal (2024) process all levels of an adaptive finite-element mesh for phase-field fracture with staged down- and up-sampling GNNs for next-step prediction, whereas AMR-VNO operates on a block-structured AMR hierarchy and predicts directly on its active leaves. AMR-Transformer constructs input-derived multiresolution tokens from dense fluid fields and applies global self-attention (Xu et al., 2025), and APT combines radius-graph encoding with Perceiver attention to learn from dynamic, locally refined subsurface meshes (Ju et al., 2026). AMR-VNO instead keeps the solver's levels, face directions, and parent-child links.

Learned multigrid and octree networks. Learned multigrid optimizes transfer for numerical convergence (Katrutsa et al., 2017; Greenfeld et al., 2019; Luz et al., 2020); MgNet links multigrid components to CNNs (He & Xu, 2019). AMR-VNO targets prediction on solver-emitted hierarchies, and MgNO is the closest multigrid neural-operator baseline we evaluate. OctNet and O-CNN allocate features and computation over octrees (Riegler et al., 2017; Wang et al., 2017), while submanifold and generalized sparse convolutions preserve sparse support over discrete lattices (Graham & van der Maaten, 2017; Choy et al., 2019); these target geometric perception. AMR-VNO combines known ingredients, namely face-neighbor stencils, learned parent-child transfer, and weight sharing across levels, into an operator whose computational and output representation is the solver's block-AMR hierarchy, with predictions only on its active leaves.

3 AMR Benchmarks

The four benchmarks play complementary roles (Table 1): Darcy isolates architectural mechanisms and parameter-count effects, HeatTransport2D provides the 2D comparison whose advantage also holds against a fine-grid reference, Quokka provides public dynamically regridded solver data,

Table 1: Benchmarks. Hierarchies are computed without the target. Splits count cases, or trajectories for Quokka; Conduction3D adds 100 validation and 100 held-out voided-family cases and 60 held-out depth-6 cases.

Benchmark	Task	Hierarchy	Train / val. / held-out
Darcy	2D elliptic	static, 64^2 to 256^2	1,100 / 100 / 200
HeatTransport2D	2D advection-diffusion	static, 128^2 to 512^2	550 / 50 / 100
Quokka	3D turbulence	solver-regridded, 32^3 to 64^3	96 / 24 / 24
Conduction3D	3D conduction	static octree to 32^3	800 / 100 / 100

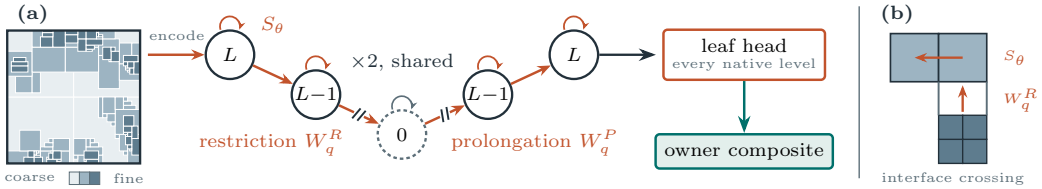

Figure 1: AMR-VNO encodes block-AMR cells, applies two shared directional V-cycles over native levels and virtual coarse levels, and predicts active leaves. The leaf-to-grid owner map is fixed, not learned. The inset shows transfer indexed by child position (Section 4).

and Conduction3D tests the V-cycle against generic message passing on the same hierarchy. In every benchmark, refinement tags and hierarchy topology are computed without access to the prediction target (*input-only*), because the target is unavailable when the model is applied. The Darcy, HeatTransport2D, and Conduction3D hierarchies are static and tagged from the input fields by the generator, while Quokka's is produced by the solver's own density-gradient tagger during the run; all are input-only in this sense.

Darcy maps a two-valued conductivity field (12 where a Gaussian random field is nonnegative and 3 elsewhere) to the zero-Dirichlet solution of $-\nabla\cdot(\kappa\nabla u) = 1$ on the unit square; conductivity interfaces alone determine its three-level hierarchy. HeatTransport2D maps initial temperature, diffusivity, and constant velocity to the periodic advection-diffusion solution at $t = 0.25$; tags follow the initial pulse along its known path and the material interfaces, without positive-time states. The public Quokka Turbulence problem (Wibking & Krumholz, 2021) evolves driven compressible hydrodynamics with the solver's density-gradient tagger, and independent forcing seeds give trajectory-disjoint splits. The model predicts future density from the current fields and horizon on the current source hierarchy, with targets restricted to it, so it does not predict future topology. Conduction3D maps anisotropic conductivity, volumetric source, and mixed boundary data to steady temperature on input-refined, balanced octrees. Its voided-composite validation family was seen during development; separate voided and depth-6 held-out sets provide further tests. Figure 2 shows one fixed validation case per benchmark, and Appendix A gives exact tagging, splits, packaging, and solver checks.

4 AMR-VNO

Figure 1 summarizes AMR-VNO. Restriction carries fine-cell information to coarser levels, and prolongation returns that coarse context to the leaves. Within a level, messages keep the direction of each face neighbor. The same smoothing and transfer weights are used at every level and cycle. We use "V-cycle" only for this fixed down-and-up schedule, not an iterative solver.

4.1 Hierarchy representation

Let V_ℓ contain cells at level ℓ. A cell i carries center $\mathbf{x}_i$, normalized physical input $\widetilde{\mathbf{a}}_i$, parent $p(i)$, and child position $q_i \in \mathcal{Q}$, and $\mathcal{C}(p)$ denotes the children of p. Let $\mathcal{D}$ denote the oriented same-level face directions. A d-dimensional cell has 2^d child positions and $2d$ face directions.

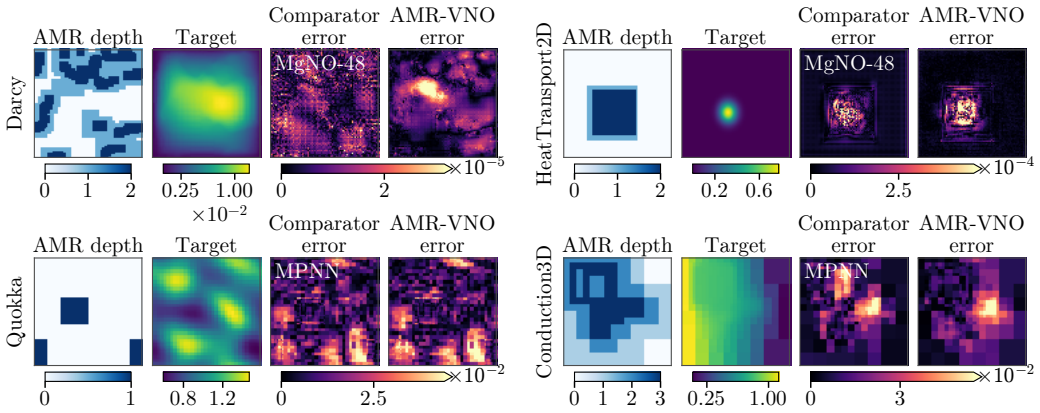

Figure 2: Fixed validation examples: relative AMR depth, target, and absolute errors of the comparator (MgNO-48 in 2D, the 12-step and 32-step hierarchy MPNN on Quokka and Conduction3D) and of AMR-VNO, all from seed 0. The two error maps of each benchmark share a color scale clipped at the 99th percentile; 3D rows are z slices, and Figure 8 adds inputs.

Each native level retains its stored patch cells, including covered parents. An *active leaf* is a cell not covered by finer cells; only active leaves produce predictions. Where a benchmark uses them, *virtual coarse levels* continue the 2:1 refinement below the solver's base level down to a single root cell; their inputs are block averages or analytic evaluations of the physical inputs, never the targets. Face edges connect cells only within a level, so communication across a coarse-fine interface follows the parent-child path rather than a cross-level face edge.

A shared encoder maps $[\mathbf{x}_i, \widetilde{\mathbf{a}}_i]$ to a d_h-dimensional hidden state. Integer levels set the schedule but are not node features, and no selected configuration encodes cell width; scale enters through the hierarchy topology and physical coordinates.

4.2 Directional smoothing and transfer

For each direction $\delta \in \mathcal{D}$, let $\mathbf{m}_{i,\delta}$ be the hidden state of the same-level face neighbor of cell i in direction δ, or zero if there is none, and let $b_{i,\delta} \in \{0, 1\}$ flag its presence. On a Cartesian lattice each direction has at most one such neighbor, so this is a masked face-stencil update. The smoother concatenates the current state with these neighbor states and flags and applies a shared residual update,

$$\boldsymbol{\xi}_i = [\mathbf{h}_i, \{\mathbf{m}_{i,\delta}, b_{i,\delta}\}_{\delta\in\mathcal{D}}], \qquad \mathbf{h}_i \leftarrow \mathbf{h}_i + S_\theta(\boldsymbol{\xi}_i). \tag{1}$$

Restriction uses a learned map indexed by child position,

$$\mathbf{r}_p = \mathrm{mean}_{c\in\mathcal{C}(p)}(\mathbf{h}_c W^R_{q_c}), \qquad \mathbf{h}_p \leftarrow \mathbf{h}_p + \mathbf{r}_p, \tag{2}$$

and prolongation adds $\mathbf{h}_{p(c)} W^P_{q_c}$ to child c. The position-shared variant ties all W^R_q to one W^R and all W^P_q to one W^P; the position-aware variant keeps one pair per child position. A cycle smooths and restricts from the finest level L to level 0, then prolongs and smooths back to L, with the same parameters at every level and in every cycle. After K cycles, the scalar head is applied only to active leaves and target normalization is inverted. For node set V, same-level face edges E, parent-child links P, and K cycles, the scheduled work is $\mathcal{O}(K(|V|+|E|+|P|))$ with graph storage $\mathcal{O}(|V|+|E|+|P|)$, and the learned parameter count is independent of hierarchy depth. Weight sharing does not imply contraction under additional cycles. Appendix A.9 gives layer, initialization, and schedule details.

4.3 Leaf prediction and compositing

Each active leaf i receives one prediction from the shared scalar head,

$$s_i = H_\theta(\mathbf{h}_i), \qquad \widehat{v}_i = \beta_v + \alpha_v g_b(\mathbf{x}_i) s_i, \tag{3}$$

where β_v and α_v invert target normalization and a known factor g_b can impose a prescribed boundary condition (Appendix A). When evaluation needs a common metric grid, a precomputed integer map $o(z)$ gives the unique leaf containing metric cell z, and that cell copies the leaf's value,

$$\widehat{v}(z) = \widehat{v}_{o(z)}. \tag{4}$$

This *leaf-to-grid owner map* has no learned parameters: there is no interpolation, nearest-neighbor search, or learned query decoder.

5 Experimental Setup

Metric and selection. The primary metric is relative L^2, $e(\widehat{u}, u) = \|\widehat{u} - u\|_2 / \|u\|_2$, computed on a common 256^2 Darcy or 512^2 HeatTransport2D grid and averaged over cases; the 3D benchmarks use volume-weighted leaf quadrature, which equals finest-grid relative L^2 for their leaf-constant targets. Development used seed 0 and validation only, and each configuration was fixed before seeds 1 and 2. Each reported 2D held-out set was generated after its checkpoints were fixed, and each checkpoint was evaluated once on it. Some of these checkpoints had also been evaluated on earlier test sets, including the packaged benchmark test splits; Appendix A.7 lists every such set and where its results are reported. The selected AMR-VNO uses two cycles, $d_h = 64$, and hidden MLP width 128, with transfer shared across child positions on Darcy, HeatTransport2D, and Conduction3D and position-specific on Quokka. Parameter counts are real scalar degrees of freedom.[1]

Baselines. Raster-input baselines (U-Net, FNO, CNO, and MgNO) receive the rasterized physical fields and, except on Quokka, refinement depth; MgNO also receives partition information. GINO-leaf receives leaf centers, leaf-sampled fields, and log cell width.

Same-hierarchy controls (the hierarchy graph U-Net, the hierarchy MPNN, and the sparse U-Net) receive AMR-VNO's cells, fields, coordinates, face graph, parent-child links, child positions, and leaf-to-grid owner map. The graph U-Net learns position-specific pooling on the same parent map. The sparse U-Net keeps AMR-VNO's native and virtual cells but uses independent sparse convolutions and fusion weights at each level (Appendix A.6.1).

MgNO-48, the 48-channel MgNO selected by a width and learning-rate search, is the headline 2D comparator. On HeatTransport2D its training loss and checkpoint selection use the leaf-to-grid owner map, which a prespecified seed-zero validation comparison chose over grid-output training. MgNO-24 is a 24-channel MgNO used in several Darcy and HeatTransport2D diagnostics. Among hierarchy MPNNs, the 32-step one is used on 2D and Conduction3D and the 12-step one on Quokka; a separate four-step Quokka MPNN serves only as a cost control.

Splits, epoch budgets, metric, and checkpoint rule are shared, but model size, optimization, and compute are not matched (Table 6). AMR-VNO evaluation uses FP16 autocast on HeatTransport2D and Conduction3D and FP32 on Darcy; dense baselines are evaluated in FP32 (Appendix A.4).

Uncertainty. Unless stated otherwise, we average each model over the three seeds for each case and bootstrap the baseline minus AMR-VNO difference across cases; the HeatTransport2D fine-reference diagnostic uses a paired Student-t interval. These intervals are conditional on the trained checkpoints and do not cover initialization, generation, split, development, or benchmark-selection uncertainty. Fine-reference, owner-map, residual, profiling, and ablation diagnostics use validation data only; the depth-transfer study uses a separate held-out stress partition.

6 Results

6.1 Does native-hierarchy learning improve accuracy?

On held-out Darcy and HeatTransport2D, AMR-VNO has the lowest mean error of all nine methods, and every paired interval against it excludes zero (Figures 3 and 4; all nine methods, with validation values, in Appendix A.5). The strongest comparator, the tuned MgNO-48, has higher error on both

[1] Each complex spectral coefficient counts twice. Counts are benchmark-specific because input channels and spatial dimension differ.

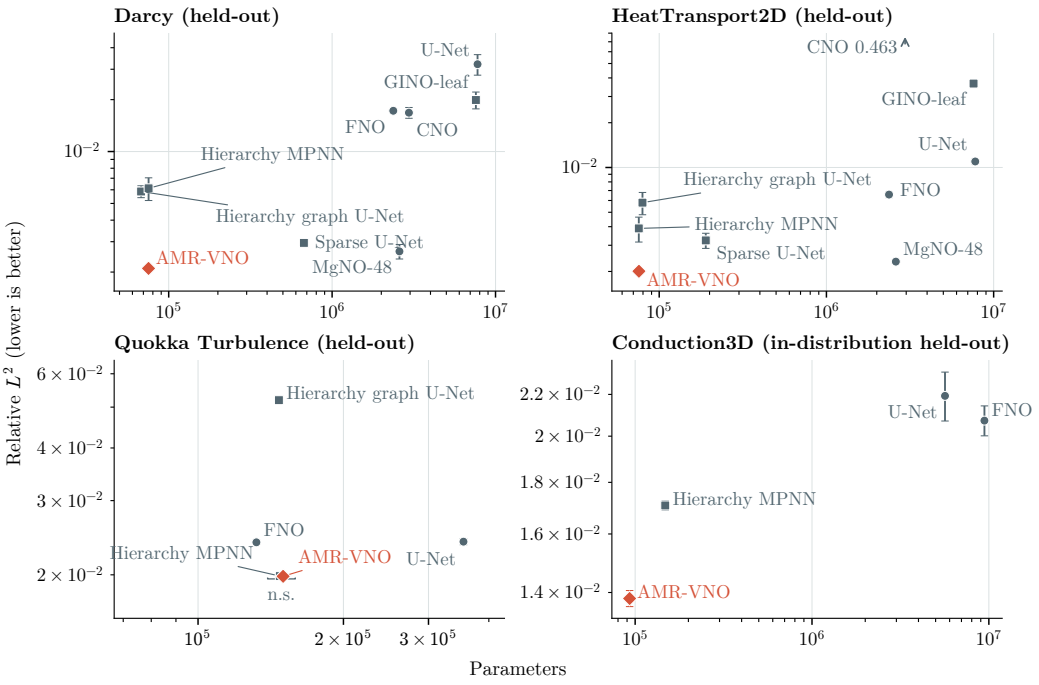

Figure 3: Held-out error against learned parameter count. Points are three-seed means with sample standard deviations; circles are raster models, squares are graph or leaf models, and the orange diamond is AMR-VNO. Lower left is better, and errors are not comparable across panels. The 2D panels show all nine methods, and Conduction3D shows its in-distribution split. On Quokka, n.s. (not significant) marks AMR-VNO and the hierarchy MPNN, whose paired 95% interval includes zero. The HeatTransport2D CNO, unstable under its bounded pinned configuration, is clipped at the panel top. FNO's complex Fourier weights count as two parameters each. Significance is in Figure 4 and the appendix tables.

benchmarks while using about 34 times as many parameters. Two caveats apply: on Darcy validation AMR-VNO and MgNO-48 are not statistically distinguishable, and on one earlier HeatTransport2D test set the interval against MgNO-48 crosses zero (Appendix A.7); Section 7 discusses benchmark selection and the single generated set per benchmark. On an earlier held-out set without MgNO-48, AMR-VNO also has lower mean error than the hierarchy graph U-Net, the hierarchy MPNN, and MgNO-24 on both 2D benchmarks (Table 20).

Among the evaluated controls, access to the same hierarchy does not reproduce AMR-VNO's gains: the hierarchy MPNN, the hierarchy graph U-Net, and the sparse U-Net all have higher held-out error on both 2D benchmarks. The hierarchy MPNN is the closest test of the update schedule: it receives the same cells, faces, and parent-child links and has nearly the same parameter count, but updates every hierarchy node at each of its 32 steps instead of traversing the levels in a V-cycle, and its held-out error is still about 2.9 and 1.9 times AMR-VNO's on Darcy and HeatTransport2D (Table 7). This does not establish superiority over hierarchy-native architectures as a family, and the sparse U-Net's Darcy selection lies on the edge of its search grid (Appendix A.6.1). On Darcy, the gain is measured against the solver's discrete targets; against a fine reference on validation, AMR-VNO's advantage over MgNO-24 is small, and MgNO-48 was not compared there (Section 6.3).

On in-distribution Conduction3D, the 32-step hierarchy MPNN has about 24% higher error than AMR-VNO, and its paired interval excludes zero; the dense models are further behind (Figures 3 and 4). On Quokka, where the model predicts future density on the current source hierarchy, AMR-VNO and the parameter-matched 12-step hierarchy MPNN are not statistically distinguishable. The evidence there favors AMR-VNO and the hierarchy MPNN over the dense baselines and does not separate the V-cycle from message passing. After normalizing Quokka error by the realized density change, AMR-VNO remains below FNO and U-Net.

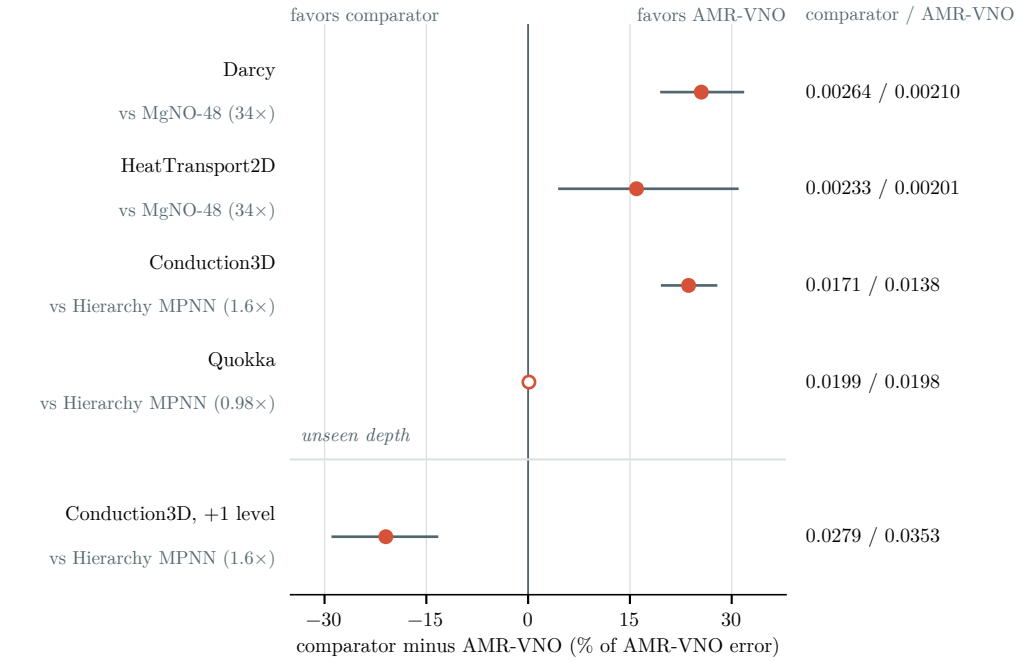

Figure 4: Paired held-out differences between each benchmark's strongest evaluated comparator (lowest held-out mean) and AMR-VNO, as a percentage of AMR-VNO's mean error. Bars are paired case-bootstrap 95% intervals divided by that error, given the checkpoints; filled points exclude zero. Every interval against the methods not plotted excludes zero in AMR-VNO's favor (appendix tables). The Darcy row is against the AMR solver's discrete targets (Section 6.3), and Quokka is not distinguishable. The last row applies the Conduction3D checkpoints, without retraining, one refinement level beyond training, where the hierarchy MPNN is better. Labels give the comparator's parameter count relative to AMR-VNO; the right column gives both held-out errors.

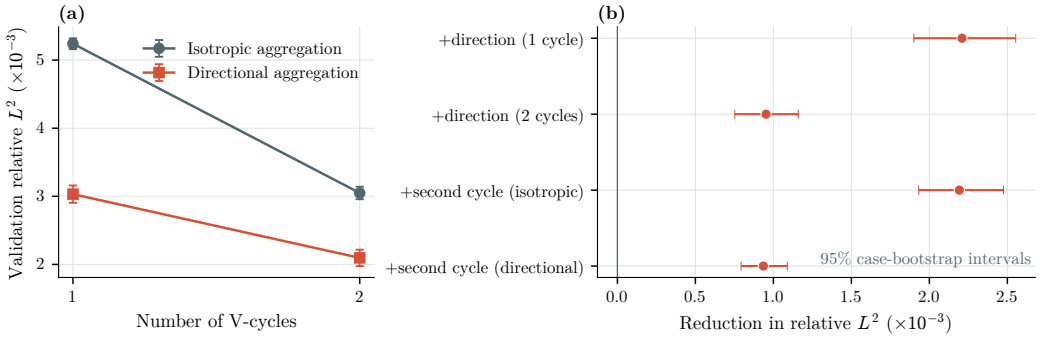

Figure 5: Darcy mechanism study within the final 75,265-parameter shared-transfer architecture. (a) Three-seed validation means with sample standard deviations. (b) Paired reductions from adding directionality or a second cycle, with 95% case-bootstrap intervals. Each mechanism improves both settings of the other; their benefits partially overlap rather than combine super-additively.

6.2 Which V-cycle mechanisms matter?

On Darcy validation, directional aggregation and a second V-cycle each reduce error with either setting of the other, and their gains partially overlap: the interaction interval excludes zero (Figure 5). As a design interpretation, directionality governs aggregation within a level, and the second cycle governs how often information traverses the hierarchy. In seed-zero Darcy validation, separately

trained variants without virtual coarse levels or without cross-level transfer have error more than an order of magnitude higher than the position-aware model they modify; the cross-level variant is smaller, so that comparison is not capacity-matched (Appendix Table 21). In a seed-zero Darcy validation control that keeps the V-cycle but replaces the adaptive hierarchy with the full finest grid, the adaptive model has lower error on 79% of cases and trains 5.4 times faster. The uniform variant still trails MgNO-24 averaged within each leaf, so this is not evidence of a uniform-grid architecture advantage (Appendix B).

6.3 Where do the advantages stop?

Numerical fidelity. The 2D targets are the AMR solver's own discrete solutions, which a surrogate is trained to reproduce, but lower error against them need not mean lower error against the underlying PDE. Against validation fine-grid references, AMR-VNO keeps its advantage over MgNO-48 on HeatTransport2D, with a paired interval that excludes zero. On Darcy validation, the solver's discrete target differs from the fine reference by about five to seven times the models' target errors, and against the reference AMR-VNO's advantage over MgNO-24 is small, about 0.5% of the reference error (Appendix A.3). Across refinement regions on validation, AMR-VNO has lower Darcy error than MgNO-48 on finest leaves and coarse-fine interfaces, and slightly higher error outside the finest level. On HeatTransport2D it has lower finest-region error but higher coarse-fine-interface error than MgNO-48 (Appendix Table 14), so the gain is not uniform across refinement regions.

Cost. Fewer parameters do not mean lower cost. On the 2D benchmarks, in the seed-zero common-hardware validation profile (Appendix A.2.2, Table 4), MgNO-48 is fastest on Darcy, AMR-VNO is fastest at batch one on HeatTransport2D, and AMR-VNO uses the most transient memory on both. Batch eight changes the HeatTransport2D latency ordering. End to end, AMR-VNO is slower than the Darcy CPU solve and faster than the HeatTransport2D solver on one deployment (Table 3). On Quokka, a four-step MPNN used only as a cost control, not the 12-step comparator, has validation accuracy similar to AMR-VNO's, and in a seed-zero profile AMR-VNO has lower resident latency and transient allocation. In a computation-only 3D sweep without targets at fixed 256^3 resolution (Figure 6), AMR-VNO has lower latency than the dense controls over the tested range and the 32-step hierarchy MPNN grows more steeply. AMR-VNO's observed latency advantage is largest when fine resolution occupies a small part of the domain, and its memory exceeds U-Net's at the largest refined fraction. At lower maximum levels in the scaling profile, the dense models are faster (Appendix Figure 9 and Table 16).

Hierarchy transfer. Sharing weights across levels does not by itself give depth invariance. Without retraining, on 60 held-out Conduction3D cases with one more refinement level, the 32-step hierarchy MPNN has lower error than AMR-VNO (0.028 against 0.035; Figure 4 and Appendix B.1), reversing the in-distribution ranking. Under a changed Darcy refinement policy, most of the validation squared error comes from the changed leaf-constant output space, and within that space AMR-VNO remains worse than the owner-projected MgNO-24, whose dense prediction is averaged within each leaf (Appendix B.4). AMR-VNO therefore depends on the hierarchy structures seen during training.

7 Limitations

Generalization. Unseen depth and a changed refinement policy degrade accuracy, so we claim no invariance to hierarchy structure. The learned cycle is also not contractive, so running more cycles at inference is not guaranteed to help.

Scope. Evidence uses three training seeds, 24 to 200 held-out cases per benchmark, and single-step prediction rather than rollouts; Quokka predicts on source leaves, not future topology. Each held-out evaluation uses one generated set per benchmark, and targets are numerical solutions. The Darcy, HeatTransport2D, and Conduction3D hierarchies are computed from the inputs rather than by solution-driven tagging (Section 3). Conduction3D uses synthetic procedural geometries, so arbitrary geometry and anisotropy remain untested. The HeatTransport2D hierarchy follows a known constant-velocity pulse path, which is favorable and not representative of general transient tagging.

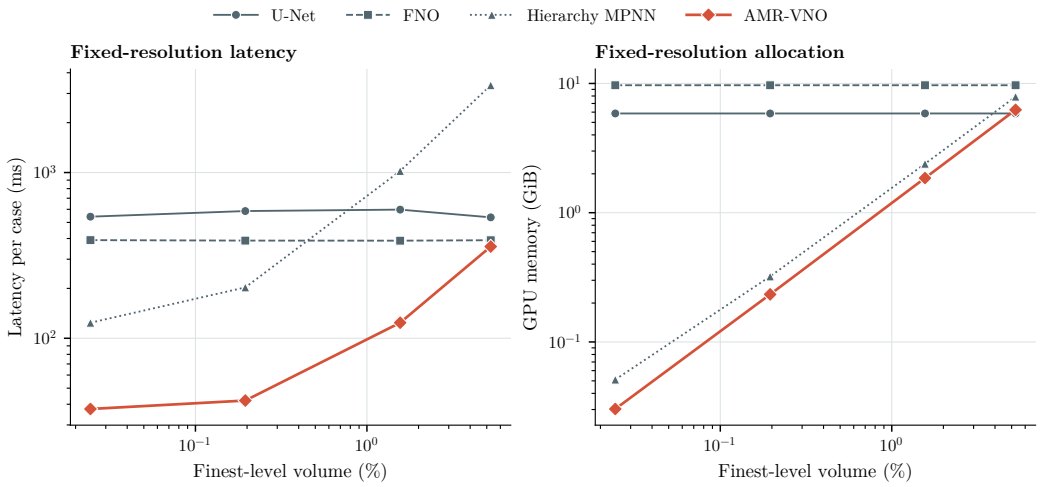

Figure 6: Conduction3D computation-only scaling at 256^3 metric resolution; no targets were generated. Centered nested refinement regions vary the realized finest-level volume from 0.0244% to 5.2734%. All models use the selected FP32 checkpoints and resident batch-one inputs on one Quadro RTX 8000. Dense controls process the same full grid at every point; hierarchy models process the changing active mesh. Memory is peak transient CUDA allocation above the resident model and input. This profile measures computation only, not accuracy or solver acceleration.

Comparison budget and selection. AMR-VNO received roughly twice MgNO's retained Darcy development hours and configurations, and more than any other baseline; these totals are lower bounds (Appendix A.6). Benchmark choice was development-selected. An earlier HeatTransport2D version failed its numerical check, and adopting the replacement required seed-zero AMR-VNO to beat all controls (Appendix A.10). This rule can bias which benchmarks are reported toward AMR-VNO. Controls are not matched in size, optimization, or compute and do not exhaust learned multiscale baselines. Transformer-based operators such as Transolver (Wu et al., 2024), GNOT (Hao et al., 2023), and UPT (Alkin et al., 2024) were not evaluated. The Darcy fine-reference comparison uses MgNO-24; MgNO-48 is untested there. The sparse U-Net's transferred 3D result is development evidence only and does not establish a sparse-convolution family optimum.

Efficiency. Cost depends on the workload. On Darcy, AMR-VNO is slower than its CPU solve and than MgNO-48 and uses more transient memory; HeatTransport2D acceleration comes from one deployment; and at the largest refined fraction of the 3D sweep, its transient memory exceeds U-Net's.

8 Conclusion

AMR-VNO shows that the solver's block-AMR hierarchy, extended where needed by virtual coarse levels, can serve directly as a neural operator's computational graph. On the evaluated held-out sets, it has the lowest mean error of nine methods on Darcy and HeatTransport2D, including the tuned MgNO-48, while using about 34 times fewer learned parameters, and it improves over the hierarchy MPNN on in-distribution Conduction3D; on Quokka, AMR-VNO and the hierarchy MPNN are not statistically distinguishable. Darcy validation ablations support directional aggregation and repeated traversal. We also release four AMR benchmarks for comparing models on solver hierarchies. Cost depends on the workload, and sharing weights across levels does not give invariance to hierarchy depth or refinement policy. AMR-VNO's lead held on test hierarchies built as in training but not one refinement level deeper on Conduction3D, and on Darcy the target discrepancy exceeds the model differences (Section 6.3). Future work includes rollouts with dynamic regridding and training across hierarchy depths so that the model can transfer between them.

AI USE STATEMENT

We used generative AI tools to refine model design and hypotheses, provide feedback on methodology and experiments, implement methods and benchmark generators, prepare the Quokka simulation data, and assist with result interpretation. The paper contains no theorems or proofs; translation, survey design, transcription, and qualitative or thematic analysis were not applicable. We also used AI for code writing and editing, suggesting experimental parameters, creating figures, identifying and summarizing related literature, formatting references, and drafting and editing the paper. The authors defined the research questions and method and made all final decisions. We reviewed all AI-assisted work: reported numbers are checked against published records by repository verifiers, code is covered by automated tests, and references were checked against bibliographic records. We take responsibility for the final content of this work.

ETHICS STATEMENT

The study uses synthetic PDE data, with no people, personal data, or private application data.

REPRODUCIBILITY STATEMENT

Code: `https://anonymous.4open.science/r/amr-vno-04EF`. Sections 3 and 5 define the data and evaluation contracts. This anonymous repository provides the generators, configurations, selected 2D AMR-VNO checkpoints, result logs, evaluation records, and independent verifiers, and its README gives one command that runs every check. Large data files and most earlier baseline checkpoint bytes are omitted from it but have deterministic generators and hash-bound evaluation records; the matched sparse U-Net package retains all 18 checkpoints used by its reported controls. Full Quokka data and all checkpoints will be released after review.

REFERENCES

Benedikt Alkin, Andreas Fürst, Simon Lucas Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=oUXiNX5KRm.

Lennart Bastian, Samuel Leventhal, Mustafa Hajij, and Tolga Birdal. Topological neural operators. *ArXiv*, abs/2606.09806, 2026. URL https://api.semanticscholar.org/CorpusID:289098442.

Marsha J. Berger and Phillip Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, 1989. URL https://api.semanticscholar.org/CorpusID:3166562.

Marsha J Berger and Joseph Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512, 1984. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(84)90073-1. URL https://www.sciencedirect.com/science/article/pii/0021999184900731.

Yadi Cao, Menglei Chai, Minchen Li, and Chenfanfu Jiang. Efficient learning of mesh-based physical simulation with bi-stride multi-scale graph neural network. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

C. Choy, JunYoung Gwak, and S. Savarese. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3070–3079, 2019. URL https://api.semanticscholar.org/CorpusID:121123422.

Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter W. Battaglia. Multiscale meshgraphnets. *ArXiv*, abs/2210.00612, 2022. URL https://api.semanticscholar.org/CorpusID:251011532.

Niklas Freymuth, Philipp Dahlinger, Tobias Würth, Simon Reisch, Luise Kärger, and Gerhard Neumann. Swarm reinforcement learning for adaptive mesh refinement. *ArXiv*, abs/2304.00818, 2023. URL https://api.semanticscholar.org/CorpusID:257913626.

Niklas Freymuth, Tobias Würth, Nicolas Schreiber, Balázs Gyenes, Andreas Boltres, Johannes Mitsch, Aleksandar Taranovic, Tai Hoang, Philipp Dahlinger, Philipp Becker, Luise Kärger, and Gerhard Neumann. AMBER: Adaptive mesh generation by iterative mesh resolution prediction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=CDEbgwJ6ET.

Hongyang Gao and Shuiwang Ji. Graph u-nets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:4948–4960, 2019. URL https://api.semanticscholar.org/CorpusID:153311899.

Benjamin Graham and L. van der Maaten. Submanifold sparse convolutional networks. *ArXiv*, abs/1706.01307, 2017. URL https://api.semanticscholar.org/CorpusID:8785126.

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.

Mingyu Han, Daniel Zhengyu Huang, Yuhan Wang, Yanshu Zhang, and Jiayi Zhou. Geometric generalization of neural operators from a kernel integral perspective, 2026. URL https://arxiv.org/abs/2602.01498.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: a general neural operator transformer for operator learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Juncai He and Jinchao Xu. MgNet: A unified framework of multigrid and convolutional neural network. *Science China Mathematics*, 62:1331–1354, 2019. doi: 10.1007/s11425-019-9547-2. URL https://api.semanticscholar.org/CorpusID:268129312.

Juncai He, Xinliang Liu, and Jinchao Xu. MgNO: Efficient parameterization of linear operators via multigrid. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8OxL034uEr.

Xin Ju, Hadrian Fung, Yuyan Zhang, Carl Jacquemyn, Matthew D Jackson, Randolph R. Settgast, Sally M Benson, and Gege Wen. Adaptive physics transformer with fused global-local attention for subsurface energy systems. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=LxNq36mefe.

Alexandr Katrutsa, Talgat Daulbaev, and I. Oseledets. Deep multigrid: learning prolongation and restriction matrices. *arXiv: Numerical Analysis*, 2017. URL https://api.semanticscholar.org/CorpusID:119599623.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: learning maps between function spaces with applications to pdes. *J. Mach. Learn. Res.*, 24(1), January 2023. ISSN 1532-4435.

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, 2020a. URL https://api.semanticscholar.org/CorpusID:224705257.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020b. Curran Associates Inc. ISBN 9781713829546.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. *J. Mach. Learn. Res.*, 24(1), January 2023a. ISSN 1532-4435.

Zongyi Li, Nikola Borislavov Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Prakash Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Anima Anandkumar. Geometry-informed neural operator for large-scale 3d PDEs. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=86dXbqT5Ua.

Songming Liu, Zhongkai Hao, Chengyang Ying, Hang Su, Ze Cheng, and Jun Zhu. NUNO: A general framework for learning parametric PDEs with non-uniform data. In *International Conference on Machine Learning*, 2023. URL https://api.semanticscholar.org/CorpusID:258967627.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3), 03 2021. ISSN ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL https://www.osti.gov/biblio/2281727.

Ilay Luz, Meirav Galun, Haggai Maron, Ronen Basri, and Irad Yavneh. Learning algebraic multigrid using graph neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Jieyuan Pei, Zhuoxuan Li, Wei Li, Haobo Zhang, jiawei jiang, and Jianwei Zheng. Hyperbolic neural operator. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=CUQwYTTNu8.

Roberto Perera and Vinamra Agrawal. Multiscale graph neural networks with adaptive mesh refinement for accelerating mesh-based simulations. *Computer Methods in Applied Mechanics and Engineering*, 429:117152, 2024. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2024.117152. URL https://www.sciencedirect.com/science/article/pii/S0045782524004080.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-NO: U-shaped neural operators. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=j3oQF9coJd.

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.

Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6620–6629, 2017. doi: 10.1109/CVPR.2017.701.

Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-CNN: Octree-based convolutional neural networks for 3D shape analysis. *ACM Transactions on Graphics*, 36(4): 1–11, 2017. doi: 10.1145/3072959.3073608. URL https://doi.org/10.1145/3072959.3073608.

Benjamin D. Wibking and Mark R. Krumholz. Quokka: A code for two-moment AMR radiation hydrodynamics on GPUs. *Monthly Notices of the Royal Astronomical Society*, 2021. URL https://api.semanticscholar.org/CorpusID:238354013.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *International Conference on Machine Learning*, pp. 53681–53705, 2024. URL https://api.semanticscholar.org/CorpusID:267411758.

Tailin Wu, Takashi Maruyama, Qingqing Zhao, Gordon Wetzstein, and Jure Leskovec. Learning controllable adaptive simulation for multi-resolution physics. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PbfgkZ2HdbE.

Zeyi Xu, Jinfan Liu, Kuang-Xu Chen, Ye Chen, Zhangli Hu, and Bingbing Ni. AMR-Transformer: Enabling efficient long-range interaction for complex neural fluid simulation. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5804–5813, 2025. URL https://api.semanticscholar.org/CorpusID:276961201.

Chenyu Zeng, Yanshu Zhang, Jiayi Zhou, Yuhan Wang, Zilin Wang, Yuhao Liu, Lei Wu, and Daniel Zhengyu Huang. Point cloud neural operator for parametric pdes on complex and variable geometries. *Computer Methods in Applied Mechanics and Engineering*, 443:118022, 2025. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2025.118022. URL https://www.sciencedirect.com/science/article/pii/S0045782525002944.

Weiqun Zhang, Ann S. Almgren, Vincent E. Beckner, John B. Bell, Johannes Blaschke, Cy Chan, Marcus S. Day, Brian Friesen, Kevin Gott, Daniel T. Graves, Max P. Katz, Andrew Myers, Tan Nguyen, Andrew J. Nonaka, Michele Rosso, Samuel Williams, and Michael Zingale. AMReX: a framework for block-structured adaptive mesh refinement. *J. Open Source Softw.*, 4:1370, 2019. URL https://api.semanticscholar.org/CorpusID:164590357.

A Reproducibility details

In tables, bold marks a benchmark's lowest mean error only when its paired interval against the next best excludes zero or, with no interval, the difference exceeds the larger seed standard deviation; grid tables bold their selected point, and parameter and cost columns are not bold.

A.1 Benchmark packaging and verification

Each benchmark stores development and held-out evaluation data in separate HDF5 containers. The 2D solver-native files store rectangular patch metadata and physical fields; the loader deterministically derives hierarchy graphs, active-leaf indices, and finest-grid owner maps. The public dense data are converted by a deterministic packager. Conduction3D instead stores direct hierarchy arrays, graph relations, leaf targets, and leaf volumes, without an owner map. Packaging checks reject non-finite values, split-size and input-pairing mismatches, invalid stored indices or level relations, nonpositive leaf volumes, and integrity-hash mismatches; the 2D path also checks exact metric-grid ownership. The independent Darcy check solves the same cell-centered discretization on a uniform 256^2 grid for eight fixed samples from each split and applies the prespecified 2% relative-L^2 gate. The HeatTransport2D gate uses an independent periodic semidiscrete solver at 1024^2; all 16 unseen pilots satisfy the fixed 0.01 cap, with maximum discrepancy 6.50×10^{-4}. Conduction3D generation checks exact octree coverage and 2:1 balance before packaging; the package audit checks stored shapes, index ranges, adjacent level relations, split seeds, finite values, positive leaf volumes summing to one, solver residual and energy imbalance below 10^{-9}, and package hashes. Every reported held-out evaluation set was generated from a manifest fixed while its target paths were absent and was audited before model evaluation.

Benchmark construction. Darcy uses native resolutions 64^2, 128^2, and 256^2, refinement ratio two, coefficient-interface tags, and a four-cell buffer. Its mean active-leaf counts are 19,935.08 on development and 19,816.06 on the packaged test split. HeatTransport2D uses 128^2, 256^2, and 512^2 levels. At $t = 0$, nine points along the known constant-velocity trajectory define the pulse corridor; relative-temperature thresholds $(0.005, 0.03)$, material-interface thresholds $(0.08, 0.25)$, and a three-cell periodic buffer define the nested masks. Mean active-leaf fractions are 22.77% on development and 22.48% on the packaged test split. Quokka retains the official density-gradient tagging at refinement ratio two. Conduction3D refines conductivity and source interfaces through level 5 and balances adjacent leaves to a one-level difference. Its procedural layered-via and finned-inclusion families provide training, validation, and in-distribution held-out cases, and a voided-composite family, seen during development, provides the voided-family validation and held-out splits. This

Table 2: Quokka Turbulence relative L^2. Mean and sample standard deviation are over seeds 0 to 2. Evaluation uses 24 trajectory-disjoint cases. Complex FNO weights count twice. Paired 95% intervals of method minus AMR-VNO: U-Net $[0.003743, 0.004539]$, FNO $[0.003709, 0.004370]$, hierarchy graph U-Net $[0.030824, 0.033469]$, hierarchy MPNN $[-0.000054, 0.000099]$.

Model	Parameters	Validation	Evaluation
U-Net	354,445	0.023157 ± 0.000307	0.023972 ± 0.000406
FNO	132,081	0.023227 ± 0.000300	0.023886 ± 0.000250
Hierarchy graph U-Net	147,041	0.051659 ± 0.000008	0.051941 ± 0.000106
Hierarchy MPNN (12 steps)	147,681	0.019378 ± 0.000030	0.019863 ± 0.000023
AMR-VNO	150,017	0.019340 ± 0.000045	0.019840 ± 0.000047

public synthetic benchmark contains no proprietary geometry, material table, parameter range, or data.

AMR-VNO benchmark instantiations. All selected instantiations use two cycles, state width $d_h = 64$, and hidden MLP width 128. Darcy and HeatTransport2D owner-composite onto their metric grids; Quokka and Conduction3D are evaluated directly on leaves. Darcy uses native levels 6 to 8 and block-averaged virtual coarse levels down to level 0; HeatTransport2D uses native levels 7 to 9, the same virtual-ancestor construction, and periodic same-level wrap edges. Quokka retains its solver-emitted octree levels 0 and 1. Conduction3D uses native levels 2 to 5 and evaluates analytic inputs at the centers of virtual coarse cells on levels 0 and 1. The 2D and Conduction3D models tie transfer across child positions; Quokka uses position-aware octree transfer. Physical inputs are standardized over training hierarchy nodes and coordinates remain unstandardized. Darcy uses $g_b(\mathbf{x}) = 16x(1-x)y(1-y)$ in Eq. (3); the other benchmarks set $g_b = 1$. Darcy and HeatTransport2D use zero-centered, maximum-magnitude target scaling, Conduction3D uses volume-weighted target standardization, and Quokka predicts a standardized density change added to the source density.

Quokka uses public source revision `0adb99a`, the pinned CPU solver executable, upstream inputs, and 120 development seeds. A separate record binds all 15 checkpoint audits and gates the generation of seeds 30000 to 30023. A new generator then writes one common package, and a target-only process audits it before model access. Every checkpoint evaluator writes an exclusive started marker before data access and atomically replaces it with the retained final evaluation record; a separate process reproduces the three-seed summary and paired intervals.

A.2 Public turbulence diagnostics

Covered Quokka coarse inputs are deterministic means of their children to numerical precision. An exact source-level channel does not improve either dense baseline, so Table 2 reports the stronger variants without it. AMR-VNO has small but nonzero mass discrepancy because neither its architecture nor loss enforces conservation. Four prespecified refinement strata show AMR-VNO ahead of FNO and U-Net throughout, yet their gap does not increase with source refined fraction; we make no scaling-accuracy claim from this test. The 24 evaluation horizons span 0.050 to 0.052 time units, and persistence has mean relative L^2 of 0.066. Relative to each case's realized density change, mean errors are 0.302 for AMR-VNO, 0.364 for FNO, and 0.365 for U-Net. AMR-VNO's mean absolute relative mass discrepancy is 2.72×10^{-4}. The solver's maximum source-normalized mass drift is 9.92×10^{-4}; because this uses a different denominator and aggregation, we do not compare the two mass values. These are post hoc descriptive statistics from immutable evaluation records and require no new inference, checkpoint loading, HDF5 access, or model selection.

A.2.1 Matched validation efficiency profile

Table 3 compares AMR-VNO with a cost-aware, parameter-matched four-step hierarchy MPNN that consumes the same supplied hierarchy. A fixed seed-zero frontier evaluates 4, 8, and 12 MPNN steps. The lowest-cost candidate retaining the observed accuracy, four steps, was then fixed before replication on seeds 1 and 2. Its profile alternates model order with seed-zero AMR-VNO on one Quadro RTX 8000. Each profile uses three warmups, then ten resident-input and five materialized-hierarchy repeats

Table 3: Efficiency diagnostics. Panel A is a Quokka validation control: its validation error is the mean over seeds 1 and 2, because seed 0 selected the MPNN step count, so the AMR-VNO value differs slightly from the three-seed value in Table 2; batch-one FP32 timing and transient allocation use seed 0. Materialized includes model-input construction from an already supplied hierarchy. Panel B reports complete benchmark-specific paths, including input-derived hierarchy construction and output materialization. Darcy entries are means and HeatTransport2D entries are medians under their own protocols; ratios are interpretable within rows only. The common-hardware 2D profile is in Table 4.

A. Quokka model-pipeline profile

Model	Validation rel. L^2	Resident ms	Resident MiB	Materialized ms / MiB
Hierarchy MPNN, 4 steps	0.019327 ± 0.000043	28.44	587.1	32.84 / 597.3
AMR-VNO, 2 cycles	0.019338 ± 0.000064	17.71	325.2	24.05 / 348.8

B. Complete benchmark-specific pipeline

Benchmark	Solver (ms)	Surrogate (ms)	Surrogate / solver	Outcome
Darcy	66.59	99.35	1.492	Slower
HeatTransport2D	6321.15	757.87	0.120	Faster

Table 4: Seed-zero validation profile on a common Quadro RTX 8000. Each cell is median milliseconds per sample / peak transient MiB. B1 and B8 denote batch sizes one and eight. The differing rankings across benchmarks and batches do not support a universal speed or memory claim.

A. Darcy

Model	B1 res.	B1 mat.	B8 res.	B8 mat.
Sparse U-Net	117.47 / 74.6	121.77 / 85.5	20.36 / 384.3	23.32 / 454.8
MgNO-48	30.49 / 56.0	30.70 / 56.5	8.51 / 448.2	8.61 / 452.2
AMR-VNO	81.04 / 156.5	89.60 / 164.3	16.78 / 783.1	19.42 / 831.5

B. HeatTransport2D

Model	B1 res.	B1 mat.	B8 res.	B8 mat.
Sparse U-Net	136.77 / 101.5	143.30 / 135.2	30.18 / 606.7	40.01 / 822.8
MgNO-48	142.15 / 224.9	145.39 / 230.9	45.07 / 1792.0	46.32 / 1840.0
AMR-VNO	99.82 / 394.4	112.30 / 416.6	36.66 / 2341.6	41.36 / 2486.3

for each of 24 validation cases. Latency is the pooled median; memory is the maximum transient CUDA allocation above the resident model and input. Clock locking was unavailable. The frontier is a cost-sensitivity diagnostic and does not alter the held-out comparison. Table 3 reports the resulting accuracy and cost, together with the complete benchmark-specific pipeline timings.

A.2.2 Common-hardware 2D profile

The common 2D profile in Table 4 uses three complete warmup passes and five measured passes over 24 fixed validation cases. Resident timing begins from model-ready accelerator inputs. Materialized timing starts from loaded CPU examples and includes collation, host-to-device transfer, inference, and output projection where required. Each accuracy pass reproduces the corresponding independent reload audit before timing is accepted. The profile measures model pipelines, not data generation or solver execution.

Table 5: Mean relative-L^2 diagnostics. "Target" is the packaged AMR target and "Ref." is a finer discretization. The 2D references are independently implemented; Conduction3D uses the same finite-volume implementation on a uniform mesh. Prediction values average three fixed seeds; HeatTransport2D diagnostic inference is FP32 for both models. For HeatTransport2D, MgNO-24 has Comparator/Ref. 0.003610, and the MgNO-48 minus AMR-VNO paired Student-t 95% interval is $[0.000204, 0.000834]$. Darcy has Target/Ref. 0.014108. Conduction3D standard deviations are over seed-level means, and its hierarchy MPNN uses 32 steps.

A. Independent-solver 2D fine-grid references

Benchmark	Comparator	Comparator/Ref.	AMR-VNO/Ref.
Darcy	MgNO-24	0.014372	0.014295
HeatTransport2D	MgNO-48	0.002523	**0.002004**

B. Conduction3D uniform-reference diagnostics

Quantity	ID validation	Voided-family validation
	Resolved-cell metric	
Target/Ref.	0.009089	0.008070
U-Net/Ref.	0.025982 ± 0.000288	0.025632 ± 0.000367
FNO/Ref.	0.023052 ± 0.000920	0.022438 ± 0.000736
Hierarchy MPNN/Ref.	0.020019 ± 0.000238	0.020072 ± 0.000404
AMR-VNO/Ref.	$\mathbf{0.018298 \pm 0.000348}$	$\mathbf{0.017657 \pm 0.000279}$
	Full level-5-grid metric	
Target/Ref.	0.070254	0.060282
Leaf-constant projection floor	0.069618	0.059700
U-Net/Ref.	0.074491 ± 0.000257	0.064729 ± 0.000239
FNO/Ref.	0.073218 ± 0.000356	0.063689 ± 0.000311
Hierarchy MPNN/Ref.	0.072940 ± 0.000085	0.063337 ± 0.000144
AMR-VNO/Ref.	$\mathbf{0.072478 \pm 0.000120}$	$\mathbf{0.062510 \pm 0.000111}$

A.3 Fine-grid numerical-reference diagnostics

For all 100 Darcy and 50 HeatTransport2D validation samples, we score the packaged target and the predictions of the three selected seeds against independent-solver finer discretizations. Separately, we score all 100 Conduction3D in-distribution validation and 100 development-visible voided-family cases against same-solver uniform level-5 references from the finite-volume implementation used for the adaptive targets. Every 3D prediction is volume-averaged onto the native leaves before applying the common leaf quadrature. The full-grid metric instead paints sparse leaf constants onto the 32^3 grid and leaves dense predictions at their native output resolution. Table 5 separates surrogate error from numerical-target discrepancy. Darcy's target-to-reference discrepancy dominates the model-to-target error, whereas HeatTransport2D's model error dominates its smaller target discrepancy. The Conduction3D ranking persists against its uniform reference on both development splits and both metrics; paired intervals, available for the full-grid metric against the hierarchy MPNN, favor AMR-VNO. On the full grid, however, every error is dominated by variation below the resolution of coarse native leaves. These development-only diagnostics quantify case variation rather than optimization uncertainty and are not continuum-error claims.

For Darcy, the fine-grid numerical-reference difference between MgNO-24 and AMR-VNO is small. The much larger packaged-target reduction therefore measures emulation of the released discrete solver, not an improvement against the fine-grid numerical reference of comparable magnitude.

A.4 Training configurations

Table 6 makes the family-specific optimization choices explicit. Within each 2D benchmark, all methods nevertheless share the data split, 500-epoch and effective-batch-eight budgets, validation cadence, checkpoint rule, and final metric, as described in Section 5; the 3D budgets follow.

Table 6: Fixed optimization configurations. "Truncated rel. H^1" retains the full Fourier relative-L^2 term and applies derivative weighting only to component frequencies strictly below 20. Cosine denotes cosine annealing over 500 epochs; the MgNO OneCycle schedule is the pinned vendor implementation.

A. Training objective

Model	Objective
GINO-leaf	Mean rel. $L^{2\dagger}$
U-Net	Mean rel. L^2
FNO	Mean rel. L^2
CNO	Batch-relative L^1
MgNO	Truncated rel. H^1
AMR-VNO	Truncated rel. H^1

B. Optimization

Model	Optimizer	LR	Weight decay	Schedule
GINO-leaf	Adam	10^{-3}	0	Cosine
U-Net	Adam	10^{-3}	10^{-4}	Cosine
FNO	Adam	10^{-3}	10^{-4}	Cosine
CNO	AdamW	10^{-3}	10^{-6}	StepLR$(10, 0.98)$
MgNO	Adam	3×10^{-4}	10^{-4}	OneCycle
AMR-VNO	Adam	10^{-3}	0	Cosine

Three-dimensional protocols. Quokka uses 50 epochs, physical batch one, AdamW, cosine annealing, and a volume-weighted density-change loss. Hierarchy models consume the exact source hierarchy, while dense controls consume the full 64^3 raster. Conduction3D uses 50-epoch, three-seed AMR-VNO, 32-step hierarchy MPNN, FNO, and nonlinear 3D U-Net runs; its dense controls consume exact 32^3 leaf-constant fields and depth. The voided family was used during development and is therefore not an unseen-distribution shift. A separate sparse U-Net control transfers the HeatTransport2D-selected width and learning rate to Conduction3D without 3D-specific tuning and is used only as development evidence.

MPNN depths. Step counts are protocol-specific rather than jointly tuned. The HeatTransport2D same-hierarchy screen selected 32 steps from 8, 16, and 32, and that depth was then used for the Darcy control; Conduction3D uses its prespecified 32-step comparator; and the Quokka held-out comparator was fixed at 12 steps. A later validation-only Quokka efficiency frontier selected four steps as the lowest-cost depth retaining its observed accuracy.

Selection mechanics. Each selected checkpoint was independently reloaded, and its validation error reproduced, before any evaluation. Architectures are fixed separately within each benchmark, and Quokka uses position-specific rather than tied transfer, so parameter counts are benchmark-specific.

†HeatTransport2D uses the prespecified truncated relative-H^1 objective for GINO-leaf; its optimizer and schedule are unchanged. All other rows apply to the two 2D solver-native benchmarks. Precisely, truncated relative H^1 is relative L^2 after an orthonormal FFT and multiplication by $w(k_x, k_y) = \sqrt{1 + \bar{k}_x^2 + \bar{k}_y^2}$, where $\bar{k} = |k|\mathbf{1}_{\{|k|<20\}}$ componentwise; high-frequency coefficients therefore retain unit weight rather than being discarded. The same-hierarchy graph U-Net and MPNN use AMR-VNO's truncated relative-H^1 objective and Adam, learning rate 10^{-3}, zero weight decay, and cosine schedule. MgNO shares that objective but retains its official optimization. HNO-leaf, a leaf-output adaptation of the Hyperbolic Neural Operator (Pei et al., 2026), uses truncated relative H^1, AdamW with learning rate 5×10^{-4} and weight decay 10^{-5}, and cosine annealing. Conduction3D and Quokka use 50 epochs, physical batch one, AdamW with learning rate 10^{-3} and weight decay 10^{-4}, cosine annealing, and a volume-weighted standardized-target MSE. Validation occurs every five epochs and selects minimum physical-unit relative L^2 for every model. AMR-VNO training on

Table 7: Held-out 2D relative-L^2 results for all nine fixed methods. Mean and sample standard deviation are over seeds 0 to 2; held-out targets (200 Darcy and 100 HeatTransport2D cases) were generated after the checkpoints were fixed. Every paired case-bootstrap 95% interval of method minus AMR-VNO excludes zero; for MgNO-48 they are $[0.000409, 0.000669]$ on Darcy and $[0.000089, 0.000623]$ on HeatTransport2D. Complex weights count twice. The small held-out HeatTransport2D MgNO-48 deviation reflects close seed means of 0.002327, 0.002336, and 0.002323.

A. Darcy

Model	Parameters	Validation	Held-out
GINO-leaf	7,582,227	0.017331 ± 0.001503	0.019902 ± 0.002227
U-Net	7,762,753	0.023107 ± 0.001893	0.032094 ± 0.004373
FNO	2,368,033	0.017302 ± 0.000145	0.017231 ± 0.000402
CNO	2,951,825	0.014380 ± 0.000242	0.016798 ± 0.001208
Hierarchy graph U-Net	67,777	0.005636 ± 0.000293	0.005865 ± 0.000463
Hierarchy MPNN	75,328	0.005960 ± 0.001200	0.006120 ± 0.000920
Sparse U-Net	672,001	0.003018 ± 0.000134	0.002952 ± 0.000094
MgNO-48	2,583,024	0.002112 ± 0.000082	0.002636 ± 0.000255
AMR-VNO	75,265	0.002096 ± 0.000120	$\mathbf{0.002100 \pm 0.000032}$

B. HeatTransport2D

Model	Parameters	Validation	Held-out
GINO-leaf	7,582,854	0.035975 ± 0.000149	0.036663 ± 0.000190
U-Net	7,763,617	0.010606 ± 0.000374	0.010974 ± 0.000433
FNO	2,368,129	0.007408 ± 0.000290	0.006575 ± 0.000301
CNO	2,953,553	0.471878 ± 0.115244	0.462995 ± 0.102250
Hierarchy graph U-Net	79,425	0.005530 ± 0.000966	0.005803 ± 0.000990
Hierarchy MPNN	75,505	0.003874 ± 0.000733	0.003895 ± 0.000742
Sparse U-Net	189,889	0.003207 ± 0.000352	0.003235 ± 0.000377
MgNO-48	2,600,772	0.002500 ± 0.000085	0.002329 ± 0.000006
AMR-VNO	75,649	$\mathbf{0.002019 \pm 0.000026}$	$\mathbf{0.002008 \pm 0.000030}$

Darcy, HeatTransport2D, and Conduction3D uses CUDA FP16 autocast with gradient scaling. Dense baseline training is FP32 except for the Conduction3D U-Net, which uses the same mixed-precision mechanism; Quokka training and evaluation are FP32 for every model. AMR-VNO evaluation uses FP16 autocast on HeatTransport2D and Conduction3D but FP32 on Darcy; other dense evaluation is FP32.

The classic CNO configuration is unstable on HeatTransport2D, so its large three-seed error in Table 7 is a bounded implementation result rather than evidence against the architecture family. Removing batch normalization briefly improves validation error before divergence, but remains noncompetitive and does not justify a broader rescue sweep. The same pinned implementation is stable on Darcy.

A.5 Broad solver-native baseline matrix

Table 7 reports every method in the reported held-out 2D evaluation, together with its validation error. The checkpoint matrix of 54 validation-selected checkpoints (nine methods, three seeds, two benchmarks) was fixed before inputs and targets were generated, targets were generated only after an independent review, and every checkpoint was evaluated once on this target set; all 54 evaluation records are retained. Earlier held-out sets on which some of these checkpoints were also evaluated are reported in Appendix A.7. This evaluation is separate from the same-information 2D control suite in Table 20, which uses its own held-out set.

Table 8: Retained Darcy development lower bounds. "Configs." counts unique seed-stripped scientific configurations across completed and excluded records; "Runs" counts unique completed checkpoint hashes. Hours include all records with measured elapsed time. Parameter ranges count stored trainable tensors, before complex weights are counted twice.

A. Headline families

Model	Configs.	Runs	Hours	Parameter range
GINO-leaf	5	7	28.8	909,907 to 3,977,747
U-Net	3	6	7.3	7,762,753 to 24,945,811
FNO	8	10	9.1	42,695 to 1,188,385
CNO	3	5	13.6	2,951,249 to 2,951,825
MgNO	19	25	72.6	73,504 to 2,583,024
AMR-VNO	36	30	160.8	56,321 to 668,161

B. Additional hierarchy controls

Model	Configs.	Runs	Hours	Parameter range
Sparse U-Net	7	9	58.9	169,089 to 672,001
Hierarchy graph U-Net	1	3	7.1	67,777
Hierarchy MPNN	4	4	126.9	75,328 to 101,857
BSMS-GNN adaptation	2	2	18.5	1,917,313 to 2,247,809

A.6 Development-budget accounting

The proposed family received substantially more development than any baseline. Table 8 reports a machine-audited retained-evidence lower bound. The path-sanitized aggregate includes unpublished and withdrawn development branches, while the table reports only Darcy families. Copied records are deduplicated, completed runs are identified by checkpoint hash, and failed, incomplete, or ineligible records contribute their scientific configuration and elapsed time whenever those fields are available. The full retained record contains 168 completed checkpoints, 128 distinct configurations, 30 excluded records, and 1,528.9 recorded wall-hours; 14 events lack recoverable elapsed time. Historical work without a retained record remains uncounted. Hours are elapsed single-accelerator training time, not normalized compute. Baseline work was deliberately bounded to upstream recipes, task adapters, and small stability or capacity checks; AMR-VNO work included architecture development, robustness experiments, and causal controls. We therefore claim a controlled comparison of final procedures, not equal tuning budgets.

The separately attested sparse U-Net protocol retains 18 completed training runs across Darcy, HeatTransport2D, and transferred Conduction3D, totaling 142.8 single-accelerator hours. This count excludes audits, profiling, preprocessing, and pilot runs before the final audit and is therefore another lower bound.

A separate, prespecified MgNO screen covers three widths and three learning rates on each 2D benchmark. Table 9 retains every outcome and the HeatTransport2D owner-map-trained continuation. An earlier pass over the HeatTransport2D grid stopped before selection after infrastructure failures. The grid was completed before selection by re-auditing the affected checkpoints with a corrected auditor and training the remaining prespecified configuration. Every selected checkpoint passes an independent reload audit, and no run accesses held-out data.

We fixed the Darcy nine-point seed-zero grid in Table 9 (Panel A) before observing its results. The selected 48-channel point reverses the seed-zero validation ranking; its three-seed validation comparison is in Table 7, and its descriptive interval is given below. This validation selection opened no targets; the separate held-out evaluation of the selected checkpoints is reported in Table 7.

For tuned Darcy, the MgNO-minus-AMR-VNO mean validation difference is 1.63×10^{-5}, with descriptive hierarchical 95% interval $[-2.22, 2.70] \times 10^{-4}$; this interval resamples both seeds and cases and is descriptive with only three seeds.

Table 9: MgNO validation relative-L^2. Panels A and B are prespecified seed-zero grids; every finite entry passed an independent reload audit. Panel C reports the selected three-seed continuations; †owner-map-trained. No row accessed held-out data.

A. Darcy seed-zero grid			
Channels	10^{-4}	3×10^{-4}	10^{-3}
24	0.005007	0.003009	Nonfinite
32	0.004308	0.002752	Nonfinite
48	0.003254	**0.002033**	Nonfinite
B. HeatTransport2D seed-zero grid			
Channels	10^{-4}	3×10^{-4}	10^{-3}
24	0.007215	0.003921	Nonfinite
32	0.004941	0.002794	Nonfinite
48	0.003364	**0.002616**	Nonfinite
C. Selected three-seed continuations			
Benchmark and model	Seed zero	Three seeds	Parameters
Darcy MgNO-48	0.002033	0.002112 ± 0.000082	2,583,024
Darcy AMR-VNO	0.002229	0.002096 ± 0.000120	75,265
HeatTransport2D MgNO-48†	0.002588	0.002500 ± 0.000085	2,600,772
HeatTransport2D AMR-VNO	0.001992	$\mathbf{0.002019 \pm 0.000026}$	75,649

Table 10: Sparse U-Net seed-zero validation relative L^2. Each finite entry passed an independent checkpoint reload audit. "Not evaluated" denotes configurations outside the retained grid, not failed runs.

Benchmark	Width	3×10^{-4}	10^{-3}	3×10^{-3}
Darcy	32	0.004893	0.004142	Not evaluated
Darcy	48	0.004211	0.003734	0.003617
Darcy	64	Not evaluated	0.003463	**0.003072**
HeatTransport2D	32	0.003872	**0.002987**	Not evaluated
HeatTransport2D	48	0.003099	0.002995	Not evaluated

A.6.1 Sparse U-Net control

The sparse U-Net of Section 5 has separate convolutions and fusion weights at each level. Table 10 reports every retained seed-zero candidate. The Darcy base-grid winner triggered a prespecified three-point boundary extension; the final selection remains on the largest evaluated width and learning rate. HeatTransport2D did not trigger extension. Seeds 1 and 2 were run only after selection. The selected three-seed checkpoints were then fixed and included in the reported held-out 2D evaluation of Table 7. An earlier held-out evaluation of the same checkpoints against AMR-VNO and MgNO-48, set C, is reported in Appendix A.7.

Figure 7 compares the selected sparse U-Net with AMR-VNO on the same Darcy validation cases: AMR-VNO has lower seed-zero validation error, fewer parameters, and lower latency but higher transient allocation.

A.7 Evaluation history

Before the reported nine-method evaluation, subsets of the same fixed checkpoints were evaluated on several earlier test sets (Table 11). Three of them are labeled A, B and C in the order they were run. Each of these three has 200 Darcy and 100 HeatTransport2D cases generated after the evaluated checkpoints were fixed. Set A evaluated the AMR-VNO, CNO, FNO, GINO-leaf and

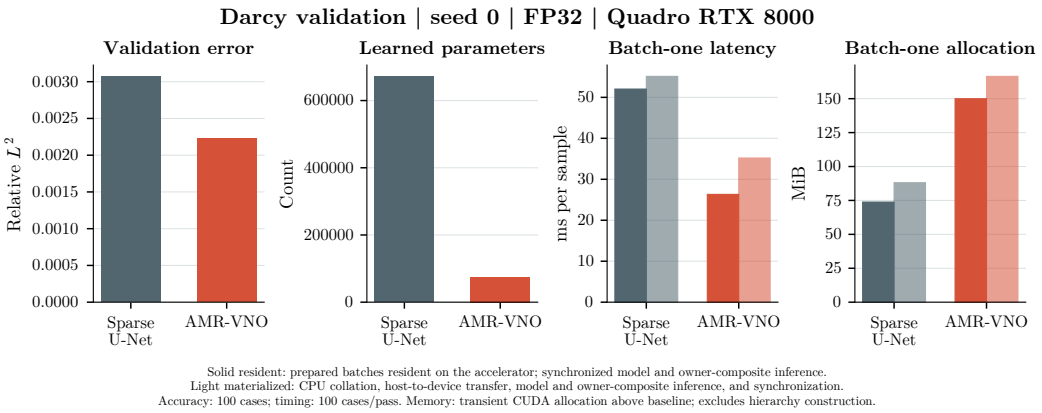

Figure 7: Same-case Darcy seed-zero validation accuracy and model-pipeline cost for the selected sparse U-Net and AMR-VNO. Solid bars use prepared accelerator-resident batches; light bars include CPU collation and host-to-device transfer. Memory is transient CUDA allocation. Measurements are FP32 on one Quadro RTX 8000 and exclude hierarchy construction and complete solver time. The timing protocol differs from those of Table 4 and Table 25B, so latencies are not comparable across them.

Table 11: Test sets on which the reported 2D checkpoints were evaluated, by date of first use. "Evaluated" lists only the nine reported methods with completed predictions. In the invalidated first set A, FNO and GINO-leaf seed 0 opened targets without predictions; in the invalidated Darcy set, MgNO-48 did; and a GINO-leaf extension of set C was interrupted without predictions. Graph U-Net and MPNN denote the hierarchy controls. "Data" is Darcy, HeatTransport2D, or both; "none" means no results are reported.

Test set	Data	Evaluated	Results
Packaged test splits	both	GINO-leaf, FNO, CNO	none
Original six-model set	both	GINO-leaf, U-Net, FNO, CNO	none
Invalidated first set A	Darcy	CNO	none
Set A	both	GINO-leaf, U-Net, FNO, CNO, AMR-VNO	Table 12
Width-32 MgNO check	Darcy	AMR-VNO	none
Same-information set	both	graph U-Net, MPNN, AMR-VNO	Table 20
Invalidated Darcy set	Darcy	AMR-VNO seed 0	none
Set B	both	MgNO-48, AMR-VNO	Table 12
Set C	both	sparse U-Net, MgNO-48, AMR-VNO	Table 12
First nine-method set	both	45 of 54 evaluations	none
Reported nine-method set	both	all nine methods	Table 7

U-Net checkpoints of Table 7, together with MgNO-24, the width-24 MgNO of Table 13, which is not part of the nine-method evaluation. Set B evaluated AMR-VNO and MgNO-48, and set C added the sparse U-Net. AMR-VNO has the lowest mean error in every set on both benchmarks (Table 12).

In set A, every method-minus-AMR-VNO paired case-bootstrap 95% interval excludes zero; for MgNO-24 the intervals are $[0.001502, 0.001821]$ on Darcy and $[0.001397, 0.001764]$ on HeatTransport2D. The MgNO-48 minus AMR-VNO interval excludes zero in set B on Darcy, $[0.000363, 0.000610]$, and HeatTransport2D, $[0.000135, 0.000452]$. In set C it excludes zero on Darcy, $[0.000375, 0.000736]$, but crosses zero on HeatTransport2D, $[-0.000082, 0.000757]$. The sparse U-Net minus AMR-VNO interval in set C excludes zero on Darcy, $[0.000760, 0.001536]$, and HeatTransport2D, $[0.000884, 0.001344]$. All evaluation records from the three sets are retained.

A first nine-method evaluation used a separate target set of the same size. It was closed after 45 of its 54 checkpoint evaluations completed and 9 failed on infrastructure, and its results are not reported (Appendix A.10). Subsets of the same checkpoints were also evaluated on other test sets:

Table 12: Earlier held-out 2D evaluations of the fixed checkpoints, in the order they were run. Values are three-seed mean and sample standard deviation of relative L^2. MgNO-24 in set A is a different checkpoint from MgNO-48.

Set	Model	Darcy	HeatTransport2D
A	GINO-leaf	0.019929 ± 0.002269	0.036361 ± 0.000151
	U-Net	0.033193 ± 0.004685	0.010767 ± 0.000280
	FNO	0.017325 ± 0.000454	0.006784 ± 0.000204
	CNO	0.016076 ± 0.001054	0.475421 ± 0.111523
	MgNO-24	0.003844 ± 0.000218	0.003624 ± 0.000247
	AMR-VNO	$\mathbf{0.002186 \pm 0.000072}$	$\mathbf{0.002057 \pm 0.000076}$
B	MgNO-48	0.002637 ± 0.000264	0.002337 ± 0.000061
	AMR-VNO	$\mathbf{0.002153 \pm 0.000068}$	$\mathbf{0.002067 \pm 0.000013}$
C	Sparse U-Net	0.003292 ± 0.000147	0.003440 ± 0.000296
	MgNO-48	0.002738 ± 0.000311	0.002671 ± 0.000029
	AMR-VNO	$\mathbf{0.002195 \pm 0.000049}$	0.002327 ± 0.000147

the packaged benchmark test splits, an original six-model set that preceded set A, a Darcy set from an invalidated first evaluation of set A, the same-information control set of Table 20, and a Darcy set for a width-32 MgNO check. A separate Darcy set was invalidated by a packaging failure. Its MgNO-48 target files were opened for all three seeds before the failure stopped them without predictions, and one AMR-VNO seed completed. Results from the two invalidated sets are not reported.

Counting only completed evaluations before the nine-method evaluation, AMR-VNO had been evaluated on seven Darcy sets (one for seed 0 only) and five HeatTransport2D sets. CNO and GINO-leaf had been evaluated on four Darcy and three HeatTransport2D sets, FNO on four sets per benchmark, and U-Net and MgNO-48 on three sets per benchmark. The sparse U-Net, the hierarchy graph U-Net and the hierarchy MPNN had each been evaluated on two sets per benchmark. In the invalidated first evaluation of set A, FNO and GINO-leaf seed 0 loaded Darcy targets but produced no predictions, and an interrupted Darcy extension of set C to GINO-leaf produced no predictions. Each checkpoint was evaluated at most once on each target set.

A.8 Darcy accuracy-capacity diagnostic

Table 13 reports a prespecified validation-only, low-parameter frontier around the 75,265-parameter AMR-VNO budget. FNO widths 6 and 8 were fixed by parameter-count calculation before optimization and then trained for the same 500 epochs as the retained FNO; the other points are independently reloaded completed runs. The 75,265-parameter AMR-VNO is more accurate than FNO and MgNO at similar parameter counts. This is a one-seed development diagnostic, not held-out evidence or a compute-matched comparison.

A.9 AMR-VNO model execution

The smoother's neighbor term is implemented as the mean over same-level neighbors in each direction, $\mathbf{m}_{i,d} = \sum_{j \in \mathcal{N}_d(i)} \mathbf{h}_j / \max(1, |\mathcal{N}_d(i)|)$, which reduces to the single neighbor state on the Cartesian lattices used here and states the aggregation if the graph contract is generalized. The encoder, smoother, and scalar head are biased two-affine-layer MLPs with GELU; transfer matrices are initialized from $\mathcal{N}(0, 0.02^2)$, affine layers use PyTorch defaults, and beyond physical-input standardization the selected model uses no batch, layer, or graph normalization. The fixed schedule encodes all cells once. Each cycle visits $\ell = L, \ldots, 1$ to smooth, mean-restrict, and add to parents; smooths V_0 when it has a same-level edge; then visits $\ell = 1, \ldots, L$ to prolong additively and smooth. The single-root V_0 update is the identity, and edgeless levels skip the smoother. No residual tolerance, stopping test, or convergence criterion is used. Appendix B reports realized schedule counts rather than labeling them as FLOPs.

Table 13: Darcy seed-zero validation relative-L^2 across the retained low-parameter capacity frontier. Complex FNO weights count twice. All points were selected on validation, independently reloaded, and accessed no held-out data.

Model	Parameters	Validation
FNO width 6	84,167	0.022660
FNO width 8	149,065	0.020618
MgNO width 8	73,504	0.010445
MgNO width 16	289,712	0.004319
MgNO width 24	648,768	0.003009
MgNO width 32	1,150,672	0.002752
AMR-VNO, 75k parameters	75,265	0.002229
AMR-VNO large	668,161	0.002032

One batch is a disjoint union of hierarchy graphs. Native and virtual cells carry a graph identifier, and all directional means and parent/child transfers are reduced only within that graph. Two complete V-cycles reuse one parameter set. The 2D losses are evaluated after exact owner compositing on the common 256^2 Darcy or 512^2 HeatTransport2D metric grid. The derived owner map is integer-valued and is not learned. Conduction3D is scored directly by leaf-volume quadrature, mathematically equal to repeating each leaf-constant value on its covered cells of a 32^3 grid.

A.10 Selection and held-out-access audit

Every final run records its configuration, split provenance, validation log, selected epoch, checkpoint hash, and per-sample errors. A separate reload audit must reproduce every validation error before a checkpoint is eligible for a reported comparison. Held-out evaluation records bind the fixed checkpoint, target bytes, and ordered sample identities, and aggregate tables are regenerated from these records. Conduction3D additionally fixed checkpoint, protocol, and source hashes before target generation. Independent recomputation reproduces the seed statistics, paired intervals, and publication gates.

Benchmark choice was development-selected. HeatTransport2D V1 failed its prespecified numerical gate; V2 used new cases and a protocol fixed before solver output, but publication still required seed-zero AMR-VNO to beat all controls. The Conduction3D warm-start branch failed its prespecified iteration and end-to-end speedup gates before multiseed evaluation and target generation; the reported protocol therefore supports no solver-acceleration claim. Infrastructure and evaluator failures, including earlier target sets with possible target access, are listed in Appendix A.7 and excluded from all reported statistics. One of these was a first nine-method evaluation of the same 54 fixed checkpoints on a separate target set. It was closed because 9 evaluations failed on infrastructure: the six CNO evaluations lacked a build prerequisite, and a training file for the three HeatTransport2D GINO-leaf evaluations was staged incorrectly. Its failure record and all 54 evaluation records are released with the nine-method evaluation, and the reported target set was generated afterwards. Reruns after those infrastructure or evaluator failures changed only the corresponding packaging or evaluator defects and reused the same unchanged checkpoints and scientific protocols. Where target access had been possible, they used new disjoint target sets. Each rerun passed a target-free rehearsal and independent recomputation.

B ADDITIONAL AUDITS AND REPRESENTATION CHECKS

Uniform representation control. A strict seed-zero control keeps the selected V-cycle, nine levels, fields, targets, objective, optimizer, and schedule fixed but replaces each adaptive hierarchy with the full 256^2 finest grid. Adaptive AMR-VNO wins 79% of paired samples, uses fewer nodes, and trains 5.44 times faster. Owner-projected MgNO-24 (three-seed 0.002793, Table 23) remains slightly better than uniform AMR-VNO (seed zero, 0.002862), so this is not evidence of a uniform-grid architecture advantage.

Paired intervals for both Conduction3D held-out splits in Table 15 favor AMR-VNO over the 32-step hierarchy MPNN. The transferred sparse U-Net is worse than AMR-VNO on both development-visible Conduction3D splits, but its configuration was selected on a different task; this comparison tests transfer of one conventional hierarchy architecture, not the sparse-convolution family.

Figure 9 and Table 16 report a fixed three-case, batch-one profile over increasing maximum level; at the finest tested level, AMR-VNO has the lowest latency and transient memory of the four models. Inputs and checkpoints are resident FP32 values on the same Quadro RTX 8000. Each model completed every case; these synthetic hierarchies test computational scaling only; no targets were generated.

An independently regenerated schedule audit over the same three level-8 hierarchies counts, per hierarchy (mean over the three), 20.85 million versus 2.61 million learned node updates and 108.50 million versus 13.56 million directed same-level messages for the MPNN versus AMR-VNO, both ratios of 8.0. The corresponding parent-child transform counts are 41.70 million versus 2.61 million, a ratio of 16.0; the MPNN also accumulates 20.85 million node contributions into graph means. These exact structural counts contextualize the latency difference but are not FLOPs or comparisons with dense FFT and convolution operations.

B.1 CONDUCTION3D DEPTH TRANSFER

Table 17 evaluates the same 12 checkpoints on 60 seed-disjoint cases whose maximum level is 6 rather than the training maximum of 5. The packages and analysis were fixed before target generation and model access; no checkpoint was retrained or adapted. AMR-VNO remains more accurate than both dense controls, but the 32-step hierarchy MPNN is more accurate on every family and is favored by a paired bootstrap analysis. This tests one new level on the same procedural families, not arbitrary depth or independent-data generalization.

A post-result protocol fixed cases 0, 9, and 19 from each family before timing the seed-zero AMR-VNO and MPNN checkpoints. Table 18 pairs each model's error on those exact cases with two inference endpoints. AMR-VNO is faster and uses less transient memory, but the MPNN is more accurate.

B.2 DIRECTIONAL V-CYCLE MECHANISM CONTROLS

The complete same-information control suite, specified separately, is reported below. Its held-out results in Table 20 use their own held-out set, distinct from the nine-method held-out evaluation in Table 7.

The removals in Table 21 show that seed-zero error rises sharply without virtual coarse levels or cross-level transfer. The selected model, which ties transfer across child positions, has slightly lower seed-zero validation error than the position-aware one (0.002229 against 0.002300; one seed, no interval; Table 25A and Table 21). In Table 22, deeper message passing narrows but does not close the validation gap. The source-checked official-family BSMS-GNN adaptation preserves upstream bi-stride selection, graph squaring, weighted transfer, multiscale skips, ReLU, and non-affine layer normalization while adding only the Darcy input, leaf-output, batching, and owner-compositing adapters. A target-free topology audit found that depth 6 is the deepest universally valid extension of the depth-5 cylinder-flow recipe on these graphs. Under the same 500-epoch exposure and validation selection, it fails its prespecified qualification gate. This follows one coverage correction from depth five to six, fixed before its run, and it receives no further tuning, additional seed, or held-out access. These validation-only controls probe attribution; they are not generalization or held-out comparisons.

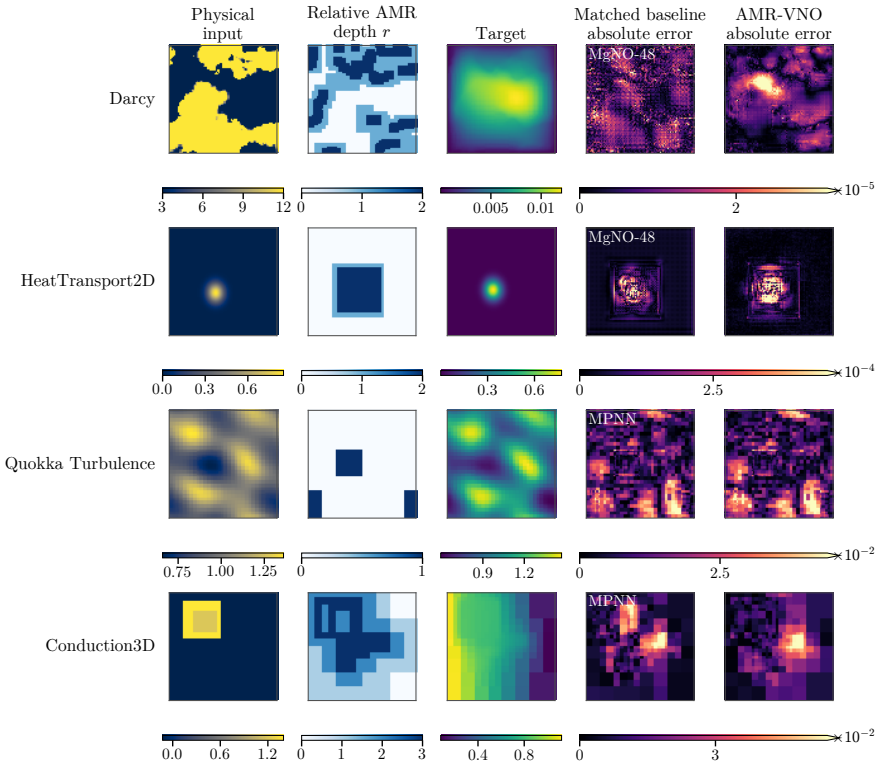

Figure 8: Validation examples. Fixed-position examples are not selected by error. Rows show a representative physical input channel (Darcy conductivity, HeatTransport2D initial temperature, Quokka density, and Conduction3D source), relative AMR depth, target, and absolute errors for the strongest comparator and AMR-VNO. The two error maps in each row share a color scale clipped at the 99th percentile. The comparator is MgNO-48 for the 2D benchmarks and the 12-step Quokka or 32-step Conduction3D hierarchy MPNN; all model outputs use seed-zero checkpoints, and 3D rows use fixed z slices. Relative depth is the leaf level minus the coarsest native level. No held-out target is shown.

Table 14: Three-seed validation relative L^2 in adaptively refined regions. Finest-leaf cells are metric cells owned by the finest native level, and the remaining metric cells lie outside the finest level. The interface comprises detected coarse-fine boundary cells followed by two four-connected dilation steps, so it overlaps both regions. Each regional error is normalized by that region's own target norm, so regions with small target values can show large relative errors. Entries are mean and sample standard deviation over seeds; no paired interval is given.

Model	Finest-leaf cells	Coarse-fine interface	Outside finest level
		Darcy	
MgNO-48	0.003385 ± 0.000018	0.002494 ± 0.000067	0.001934 ± 0.000098
AMR-VNO	$\mathbf{0.002850 \pm 0.000133}$	$\mathbf{0.002280 \pm 0.000120}$	0.001994 ± 0.000119
		HeatTransport2D	
MgNO-48	0.002394 ± 0.000092	$\mathbf{0.067478 \pm 0.001108}$	0.172426 ± 0.008770
AMR-VNO	$\mathbf{0.001925 \pm 0.000035}$	0.085686 ± 0.000863	$\mathbf{0.148624 \pm 0.005436}$

The matched final-architecture ablation is reported in Table 19. Reusing the selected weights for a third cycle is unstable because repeated transfer amplifies latent features; this diagnostic is not a

Table 15: Conduction3D volume-weighted relative-L^2. Mean and sample standard deviation are over seeds 0 to 2. The voided-family cases contain only voided-composite geometries, a family visible during development. Complex FNO weights count twice. The sparse U-Net transfers the selected HeatTransport2D configuration without 3D tuning and does not access Conduction3D held-out targets. The hierarchy MPNN uses 32 steps. Held-out paired 95% intervals of method minus AMR-VNO, in distribution: U-Net $[0.007328, 0.008944]$, FNO $[0.006216, 0.007630]$, hierarchy MPNN $[0.002702, 0.003851]$; voided family: $[0.007456, 0.009355]$, $[0.005711, 0.007495]$, $[0.002012, 0.003061]$.

A. In-distribution geometries

Model	Parameters	Validation	Held-out
U-Net	5,650,273	0.025361 ± 0.000412	0.021926 ± 0.001222
FNO	9,441,585	0.022893 ± 0.000878	0.020723 ± 0.000707
Hierarchy MPNN	147,969	0.018256 ± 0.000270	0.017075 ± 0.000178
Sparse U-Net	291,585	0.019814 ± 0.000656	Not evaluated
AMR-VNO	93,057	$\mathbf{0.015706 \pm 0.000267}$	$\mathbf{0.013810 \pm 0.000253}$

B. Voided-family cases (family visible during development)

Model	Parameters	Validation	Held-out
U-Net	5,650,273	0.025174 ± 0.000509	0.023198 ± 0.001167
FNO	9,441,585	0.022551 ± 0.000821	0.021408 ± 0.000624
Hierarchy MPNN	147,969	0.018786 ± 0.000495	0.017373 ± 0.000418
Sparse U-Net	291,585	0.018254 ± 0.000474	Not evaluated
AMR-VNO	93,057	$\mathbf{0.015682 \pm 0.000234}$	$\mathbf{0.014851 \pm 0.000519}$

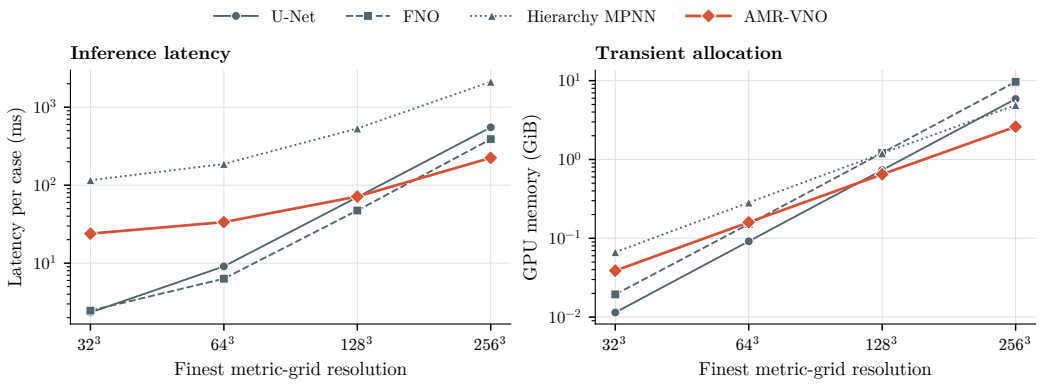

Figure 9: Conduction3D computational scaling on three fixed layered-via hierarchies per resolution. Adaptive models process native hierarchy nodes, while dense models process the full metric grid. Memory is peak transient CUDA allocation above the resident model and input. This profile excludes hierarchy construction and makes no accuracy claim.

separately trained architecture. All mechanism comparisons are validation-only and accessed no held-out data.

B.3 Output-resolution and boundary checks

Dense methods can vary within a coarse leaf whereas AMR-VNO cannot. We therefore reloaded every dense checkpoint, averaged each prediction within its exact owner cell, and rescored the resulting leaf-constant field. The dense baseline tested here, MgNO-24, remains worse after this projection (Table 23); GINO-leaf and AMR-VNO already have leaf-constant outputs. The projection was specified after training and is reported as a diagnostic, not a held-out comparison. A separate

Table 16: Conduction3D structure counts, latency in ms, and transient memory in GiB. Counts are three-hierarchy means rounded to the nearest integer. AMR-VNO and the MPNN update all hierarchy nodes; predictions are defined on leaves. The hierarchy MPNN uses 32 steps; dense methods process all voxels.

A. Structure count

Level	Leaves	Nodes	Face edges	Parent-child	Voxels
5	7,470	8,537	22,351	8,536	32,768
6	32,931	37,636	97,515	37,635	262,144
7	139,058	158,924	413,788	158,923	2,097,152
8	570,186	651,641	1,695,355	651,640	16,777,216

B. Latency (ms)

Model	Level 5	Level 6	Level 7	Level 8
U-Net	2.33	9.10	70.20	550.99
FNO	2.46	6.33	47.33	388.90
Hierarchy MPNN	115.74	186.98	533.11	2,116.74
AMR-VNO	23.89	33.59	71.46	223.94

C. Transient memory (GiB)

Model	Level 5	Level 6	Level 7	Level 8
U-Net	0.01	0.09	0.73	5.86
FNO	0.02	0.15	1.21	9.69
Hierarchy MPNN	0.07	0.28	1.20	4.89
AMR-VNO	0.04	0.16	0.65	2.60

Table 17: Conduction3D maximum-level-6 volume-weighted relative-L^2 without retraining. Mean and sample standard deviation are over the three fixed training seeds. Family columns denote layered via, finned inclusion, and voided composite; the hierarchy MPNN uses 32 steps. Paired 95% intervals of method minus AMR-VNO over all 60 cases: U-Net $[0.274509, 0.305187]$, FNO $[0.020946, 0.027555]$, hierarchy MPNN $[-0.010226, -0.004670]$.

A. Layered and finned families

Model	Layered	Finned
U-Net	0.330082 ± 0.042912	0.316660 ± 0.039499
FNO	0.057346 ± 0.001015	0.061297 ± 0.000582
Hierarchy MPNN	$\mathbf{0.023197 \pm 0.001317}$	$\mathbf{0.030420 \pm 0.001266}$
AMR-VNO	0.028176 ± 0.000648	0.038850 ± 0.001275

B. Voided family and aggregate

Model	Voided	All
U-Net	0.328843 ± 0.040837	0.325195 ± 0.040855
FNO	0.060213 ± 0.000707	0.059619 ± 0.000624
Hierarchy MPNN	$\mathbf{0.030044 \pm 0.001465}$	$\mathbf{0.027887 \pm 0.001339}$
AMR-VNO	0.038835 ± 0.001702	0.035287 ± 0.001132

prespecified control applies the same projection before MgNO's loss and checkpoint selection. This narrows but does not reverse the Darcy ranking at either tested width (MgNO-24 and MgNO-32, with 24 and 32 channels). The earlier HeatTransport2D MgNO-24 control remains single-seed because it

Table 18: Conduction3D maximum-level-6 same-case accuracy and cost on nine fixed cases. Error is mean volume-weighted relative L^2 for seed 0; latency is the median per case in ms and memory is mean transient MiB on one Quadro RTX 8000. Materialized timing starts from an already loaded hierarchy and includes CPU collation and host-to-device transfer.

Model	Error	Latency (ms)		Transient memory (MiB)	
		Resident	Materialized	Resident	Materialized
Hierarchy MPNN	0.023820	269.29	750.87	222.35	226.13
AMR-VNO	0.029804	47.58	500.32	136.48	142.99

Table 19: Darcy validation ablation within the 75,265-parameter shared-transfer architecture. Means and sample standard deviations are over seeds 0 to 2. The cycles-by-directionality difference-in-differences is 0.001256, with case-bootstrap 95% interval $[0.000970, 0.001561]$.

Variant	Relative L^2
One isotropic cycle	0.005241 ± 0.000080
One directional cycle	0.003032 ± 0.000127
Two isotropic cycles	0.003049 ± 0.000092
AMR-VNO, two directional cycles	$\mathbf{0.002096 \pm 0.000120}$

failed its continuation gate; the later selected MgNO-48 continuation also remains behind AMR-VNO. Table 23 reports these validation-only comparisons, which do not alter any held-out comparison.

The fixed Darcy boundary envelope is also not necessary for the ranking. Table 23, Panel C compares separately trained seed-zero models: AMR-VNO wins when neither method receives the envelope and when both receive it. The envelope helps AMR-VNO relative to its own no-envelope result but does not explain the method ordering.

B.4 Policy, depth, and compute controls

These Darcy controls use the released buffer-four solution as a common target. Buffer two preserves every coefficient composite, changes every hierarchy, and substantially reduces the active-leaf count. Policy-specific and mixture training do not recover the original ranking. A prespecified continuous-query successor improves the shifted-policy result but remains depth-dependent and computationally slower than MgNO. The fixed control in Table 24 instead consumes the buffer-two hierarchy and depth input and separates the exact leaf-constant projection floor from learned error within that output space. For the volume-weighted orthogonal owner projection $P_{\mathcal{H}}$ and a leaf-space prediction $\widehat{u}_{\mathcal{H}}$,

$$\frac{\|\widehat{u}_{\mathcal{H}} - u\|_V^2}{\|u\|_V^2} = \frac{\|\widehat{u}_{\mathcal{H}} - P_{\mathcal{H}}u\|_V^2}{\|u\|_V^2} + \frac{\|P_{\mathcal{H}}u - u\|_V^2}{\|u\|_V^2}. \tag{5}$$

We compute every component per case before averaging.

The buffer-two leaf-constant floor is 0.009046. As a mean over cases and seeds of the squared-error share, it accounts for 86.8% of AMR-VNO's squared error and 89.3% of owner-projected MgNO's. AMR-VNO remains slightly worse after output matching, rejecting policy invariance while showing that the changed output space causes most squared error.

Matched-budget HeatTransport2D cyclic and donor layouts reverse the physical-layout ranking: owner-projected MgNO approaches the leaf-constant floor, while both hierarchy models remain much worse. These solution-uninformative layouts are stress tests, not practical AMR policies.

Table 25 removes virtual coarsening levels and profiles resident FP32 inputs on one Quadro RTX 8000.

Monotone degradation in Table 25A diagnoses dependence on global coarsening, not unseen-depth transfer. Panel B shows that AMR-VNO has fewer parameters than MgNO-24 but is slower and

Table 20: Complete same-information 2D controls on held-out relative L^2. Means and sample standard deviations are over seeds 0 to 2. HNO-leaf was evaluated only on HeatTransport2D; both 2D hierarchy MPNNs use 32 steps.

Model	Darcy	HeatTransport2D
HNO-leaf	Not evaluated	0.016394 ± 0.006065
Hierarchy graph U-Net	0.006314 ± 0.000633	0.005327 ± 0.000948
Hierarchy MPNN	0.006615 ± 0.000848	0.003856 ± 0.000653
MgNO-24	0.003819 ± 0.000207	0.003391 ± 0.000225
AMR-VNO	$\mathbf{0.002181 \pm 0.000019}$	$\mathbf{0.001968 \pm 0.000065}$

Table 21: Seed-zero Darcy validation removals. Every row is a separately trained 500-epoch variant of the 99,841-parameter position-aware model, not the selected 75,265-parameter shared-transfer model; the cross-level removal has 67,073 parameters, so its rise is not capacity-matched.

Variant	Parameters	Validation rel. L^2
Position-aware, one V-cycle	99,841	0.002695
Position-aware, two V-cycles	99,841	0.002300
Isotropic aggregation	99,841	0.003071
Trained without virtual coarse levels	99,841	0.120797
Trained without cross-level transfer	67,073	0.162035

uses more transient memory. The released profiles separately measure the complete heterogeneous hierarchy-to-output paths. The HeatTransport2D complete-path profile uses one thread of an Intel Core i9-10980XE and the same GPU; storage and clock-control metadata were not retained.

Table 22: Additional seed-zero Darcy validation controls on the same data. The hierarchy graph U-Net and hierarchy MPNNs receive the same nodes, fields, coordinates, directional edges, parent-child edges, and owner composite as AMR-VNO; the BSMS-GNN adaptation builds its own bi-stride levels without AMR-VNO's covered parents, parent-child edges, or virtual coarse levels. Removal variants of AMR-VNO are in Table 21.

Model	Parameters	Validation rel. L^2
BSMS-GNN adaptation	2,247,809	0.041957
Hierarchy graph U-Net	67,777	0.005577
Hierarchy MPNN (four steps)	101,857	0.060895
Hierarchy MPNN (32 steps)	75,328	0.005491

Table 23: Output-space and boundary controls. Panels A and B report three-seed validation relative L^2 under exact owner projection; "Direct" is the standard dense score. Panel C reports separately trained seed-zero Darcy boundary treatments.

A. Projection applied only at evaluation

Benchmark	Model	Direct	Owner-projected
Darcy	MgNO-24	0.003000 ± 0.000026	0.002793 ± 0.000016
	AMR-VNO	$\mathbf{0.002096 \pm 0.000120}$	$\mathbf{0.002096 \pm 0.000120}$
HeatTransport2D	MgNO-24	0.003637 ± 0.000257	0.003516 ± 0.000254
	AMR-VNO	$\mathbf{0.002019 \pm 0.000026}$	$\mathbf{0.002019 \pm 0.000026}$

B. Projection applied during training

Benchmark	Model	Validation
Darcy	MgNO-24	0.002663 ± 0.000091
	MgNO-32	0.002397 ± 0.000110
	AMR-VNO	$\mathbf{0.002096 \pm 0.000120}$
HeatTransport2D	MgNO-48	0.002500 ± 0.000085
	AMR-VNO	$\mathbf{0.002019 \pm 0.000026}$

C. Darcy boundary treatment, seed zero

Model	Envelope	Parameters	Validation rel. L^2
MgNO-24	No	648,768	0.003009
	Yes	648,768	0.003201
AMR-VNO	No	75,265	0.002750
	Yes	75,265	0.002229

Table 24: Darcy validation-only buffer-two input and output control against one common buffer-four target. Mean and sample standard deviation are over three fixed seeds. "Within space" is the square root of the first squared component in Eq. 5. No held-out target was accessed.

Model	Output space	Rel. L^2	Within space
MgNO-24	Dense	0.009550 ± 0.000030	n/a
	Buffer-two owners	0.009636 ± 0.000024	0.003047 ± 0.000041
AMR-VNO	Buffer-two leaves	0.009753 ± 0.000025	0.003441 ± 0.000087

Table 25: Darcy hierarchy-depth and inference diagnostics. Panel A removes virtual coarsening levels from the fixed seed-zero AMR-VNO checkpoint. Panel B is a separate resident-input profile on the same Quadro RTX 8000, and its comparator is MgNO-24, not MgNO-48. Latency is the per-sample mean over the 100 validation cases at batch one and the 96 cases in full batches at batch eight, and peak memory is transient allocation. Its protocol differs from Table 4, so latencies are not comparable across the two tables.

A. Selected-model depth dependence

Levels	Rel. L^2	Levels	Rel. L^2
9	0.002229	5	0.159280
8	0.016971	4	0.181006
7	0.064215	3	0.261480
6	0.121997		

B. Resident-input inference

Model	Batch	Latency (ms)	Samples/s	Peak (10^6 B)
MgNO-24	1	4.11	243.3	29.3
	8	2.83	353.0	234.7
AMR-VNO	1	38.10	26.2	130.2
	8	10.70	93.5	687.0