CoRaWS: Zero-Shot Neural Warm Starts across Topology Families for Nonlinear Magnetostatics

Anonymous authors
Paper under double-blind review

Abstract

Neural network surrogates have attracted broad interest because they can support repeated physical simulations at low inference cost. However, if deployment on an unseen target topology family still requires collecting high-fidelity labels and training a model, the preparation cost may offset the gains at inference. Even when field prediction errors are low, the predicted solution may not satisfy the numerical solver's convergence criteria. Applications to unseen topology families therefore need to control data preparation costs while ensuring the accuracy of the final solution. For nonlinear magnetostatics, we propose complementary approaches to data construction and initial-guess prediction. We first develop EMTopoGen, a parametric topology generator combining primitive shapes, assembly relations, and continuous parameters. Using it, we construct EMTopoBench-54 with 54 topology families and 27,000 high-fidelity simulation samples. This benchmark supports the study of compositional generalization through training on source families and evaluation on unseen target families. We then develop CoRaWS, a neural warm-start method based on constitutive coordinate reparameterization and field-magnitude awareness. After training on source families, the method generates finite element initial guesses for unseen target topology families without target-family labels or fine-tuning, accelerating the solution process while maintaining high fidelity. Experiments on nonlinear magnetostatic problems show that CoRaWS offers advantages for FEM acceleration, indicating its potential for engineering simulation. Code will be released upon acceptance.

1 Introduction

The finite element method (FEM) obtains numerical solutions to partial differential equations (PDEs) through weak formulations and mesh discretization (Brenner & Scott, 2008), but complex geometries and material nonlinearities make repeated solves expensive. To reduce this cost, neural operators directly learn mappings from problem conditions to solutions (Li et al., 2021), with recent architectures demonstrating rapid prediction of complex physical fields (Wang et al., 2025; Luo et al., 2025; Holzschuh et al., 2025). Beyond architectural changes, spectral derivatives improve the representation of local variations (Liu et al., 2026b), while joint learning of fundamental physical knowledge improves data efficiency and out-of-distribution generalization (Ma et al., 2026). Physical structure also helps constrain predictions through energy and dissipation laws for dynamical stability (Roth et al., 2025) and PDE constraints in operator learning (Li et al., 2024). Even so, low prediction error does not automatically imply that a numerical solver's convergence criteria are satisfied, and further numerical correction may be needed (Taghikhani et al., 2026). Supervised training also requires expensive simulations to obtain field labels in advance. To reduce label requirements, geometry-based self-supervised pretraining uses unsolved geometry samples, whereas PFEM trains directly through physical equations without solution labels (Zhang et al., 2026; Wang et al., 2026). Even with these advances, any sampling, simulation, or adaptation required for a new topology family must be weighed against the gains from fast inference. Practical acceleration should therefore be measured at matched accuracy and include the additional computation required for deployment (McGreivy & Hakim, 2024).

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.

These challenges arise together in nonlinear magnetostatic computation, an important part of electric machine and magnetic device design. In particular, material saturation changes the local constitutive response that enters the FEM Newton linearization (Egger et al., 2026). The same deviation in magnetic flux density B can therefore correspond to different deviations in magnetic field strength H at different magnetization states. An average B-error can hide these differences, making it an incomplete measure of initial-guess quality. At the same time, changing component combinations and assembly relations can alter magnetic paths. PDE symmetries help reuse existing solutions or models through data augmentation (Brandstetter et al., 2022) and operator canonicalization (Shumaylov et al., 2025), but these transformations do not generally account for changes in component assembly. To handle changes in the domain itself, GINOT and GAOT improve geometric representations (Liu et al., 2026a; Wen et al., 2025), while SNI uses domain decomposition for inference on new geometries (Huang et al., 2026). However, their geometric flexibility alone does not establish that predictions will accelerate FEM on unseen topology families. Even within one topology, different excitations produce a wide range of field magnitudes and solution costs. For cases that already converge rapidly from zero, constructing a neural initial guess may cost more time than it saves.

To address these challenges, we develop EMTopoGen, a parametric topology generator, and CoRaWS, a neural warm-start method. EMTopoGen combines primitive shapes and their assembly relations under geometric and physical constraints, then samples continuous parameters. We use it to construct EMTopoBench-54, a compositional generalization benchmark with 54 topology families and 27,000 high-fidelity simulation samples, holding out entire families for evaluation. For initial-guess prediction, CoRaWS uses constitutive coordinate reparameterization: it learns fields in H coordinates and recovers B in magnetic regions through the known material relation. We analyze how this coordinate choice changes local error weights and connects to the differential reluctivity used in FEM. To handle the wide dynamic range of field values, the model combines range-aware encoding with joint supervision of B, H, and secant magnetic permeability μ. The predicted magnitude class selects between zero and neural initialization to control construction overhead. The model is trained only on source-family data from EMTopoBench-54 and requires neither target-family labels nor fine-tuning. For neural initialization, the candidate field is projected to a magnetic vector potential, after which the original FEM solver completes the solve on the same mesh under the same convergence criteria.

Our contributions are as follows:

1. **A compositional topology generation method and benchmark.** We propose EMTopoGen, a parametric topology generation method subject to geometric and physical constraints, construct EMTopoBench-54 with 54 topology families and 27,000 high-fidelity samples, and establish a topology-family holdout protocol to support the study of field prediction and solver acceleration on unseen structural combinations.
2. **A constitutive- and range-aware neural warm-start method.** We establish a framework for constitutive coordinate reparameterization, derive how the constitutive mapping and nonlinear encoding jointly change local error weights, and explain their connection to differential reluctivity in FEM. On this basis, we propose CoRaWS, which combines range-aware encoding, joint supervision of B, H, and μ heads, and recovery through the material constitutive relation to predict initial guesses without target-family labels or fine-tuning.
3. **Evaluation of acceleration across topology families at matched accuracy.** We conduct zero-shot experiments on 1,440 evaluation pairs across three topology-family splits using the same meshes and convergence criteria, and analyze the role of each component through comparisons and ablations. Including network inference and initial-guess construction, CoRaWS reduces cumulative online time by 8.41% and the total number of Newton iterations by 11.79% relative to zero initialization.

2 Related Work

2.1 Geometric Generalization and Field Prediction Benchmarks

GINOT and GAOT extend neural operators to varying computational domains through geometry-aware representations (Liu et al., 2026a; Wen et al., 2025). SNI reuses local operators through domain decomposition to solve problems on new geometries (Huang et al., 2026). Geometry-based self-supervised pretraining reduces the need for field labels by using unsolved geometry samples

(Zhang et al., 2026). For evaluation, PDEBench and The Well provide public field prediction tasks for a range of physical systems (Takamoto et al., 2022; Ohana et al., 2024). Evaluating geometric generalization also requires specifying exactly what changes between training and testing. Varying dimensions, positions, or excitations while keeping the component combination fixed constitutes parametric generalization. Topological generalization instead requires a model to handle changes in component combinations and assembly relations. For magnetic devices, changes in assembly relations can alter magnetic circuit connections and branches. We evaluate transfer across these structures by holding out entire topology families.

2.2 Physics-Guided Learning and Neural Warm Starts

Physical knowledge can enter field prediction through training objectives or network architectures. For example, PINO incorporates PDE constraints into operator learning (Li et al., 2024), stable port-Hamiltonian neural networks use energy and dissipation structures to constrain dynamics (Roth et al., 2025), and multipatch neural solvers combine geometric mappings with variational energies to solve problems on CAD domains, including magnetic field examples (von Tresckow et al., 2026). When a predicted field still requires correction by numerical iteration, reducing field error is not equivalent to saving solution time. Traditional numerical methods reduce iterations by improving the initial guess, for example by interpolating a coarse-mesh solution onto a fine mesh as a starting point for the subsequent nonlinear solve (Adler et al., 2010). Neural warm starts instead construct initial guesses from model predictions and use the original solver to complete the correction. To connect initial-guess learning with subsequent iterations, Learning to Warm-Start appends a fixed number of iterations to the predicted initial guess and trains the initialization model using a downstream objective (Sambharya et al., 2024). For specific solution tasks, Neural-Initialized Newton uses neural predictions to initialize nonlinear finite element iterations (Taghikhani et al., 2026), while NOWS constructs initial guesses for Krylov iterative solvers (Eshaghi et al., 2026). Initial guesses can also be learned without solution labels, as PFEM uses physics-based pretraining to provide FEM warm starts (Wang et al., 2026). For magnetic problems, a Monumo technical article studies predicting the magnetic vector potential of electric machines and then completing the solve with Newton iterations (Nonnenmacher & Durrande, 2026). We further examine field representations under material saturation and their effectiveness as warm starts for unseen assembly topology families. Our evaluation considers both iteration savings and initial-guess construction costs to test whether learning on source families can reduce the total time needed to meet the same convergence criteria without target-family labels or adaptation.

3 Method

We consider two-dimensional magnetostatics with single-valued, isotropic material relations and no hysteresis. EMTopoGen separates assembly templates from within-family parameters to provide training data for CoRaWS, which learns fields through constitutive coordinates and range-aware encoding, then selects zero or neural initialization from the predicted magnitude class. Source and target families are disjoint, and both the model and data-driven preprocessing are fitted only on source families, without target-family labels or fine-tuning. The magnetostatic formulation and FEM solution procedure are given in Appendix A.

3.1 EMTopoGen: Compositional Topology Generation and Benchmark Construction

Figure 1(a) varies the width at a fixed height within a two-layer staggered-shield template, whereas the bypass armature and tapered pole shoes in Figure 1(b) differ in component combinations and assembly layouts. EMTopoGen represents these variations through a parametric object tree with topology template $\tau = (\mathcal{S}, \mathcal{R}, \mathcal{D})$, comprising shapes and components $\mathcal{S}$, relational constraints $\mathcal{R}$, and parameter dependencies $\mathcal{D}$. After sampling free parameters $\mathbf{p}$, the dependency graph determines the remaining parameters to produce geometry $G_\tau(\mathbf{p})$. Fixing τ gives within-family variation, while changing its components or relations defines a new family. These families describe engineering assembly structures and need not belong to distinct mathematical homeomorphism classes.

EMTopoGen checks objects, hierarchies, and geometric relations before the downstream task assigns materials, currents, and the outer boundary and calls Maxwell to solve and export fields. Acceptance requires valid solver status, complete outputs, and valid field arrays. Additional candidates are sampled with unchanged parameter ranges, physical settings, and convergence thresholds until each family has 500 accepted samples. EMTopoBench-54 thus contains 9 engineering groups, 54 topology families, and 27,000 samples. Entire templates are held out, keeping target-family assembly structures unseen during training.

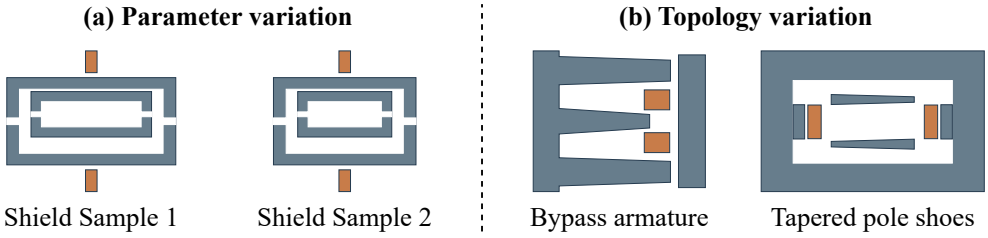


Figure 1: Two kinds of geometric variation in EMTopoGen. (a) Sample 1 and Sample 2 have the same vertical dimension but different widths, representing parameter variation within the same topology. (b) Differences between distinct topologies. Blue-gray regions represent magnetic material, orange represents coils, and white represents air or gaps.

3.2 Constitutive Coordinate Reparameterization

Near saturation, increasing H produces progressively smaller increases in B, as shown in Figure 2(a). Consequently, the same B-magnitude deviation at the three operating points in Figure 2(b) produces a much larger change in H near saturation than in the weak-field regime. We examine error in H coordinates to make supervision reflect this material dependence.

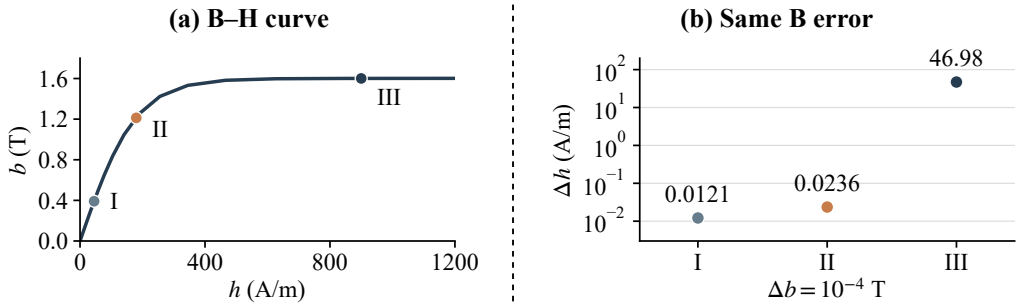


Figure 2: Effects of the same deviation in B magnitude at different material states. (a) The piecewise linear material curve used in this work, with I, II, and III corresponding to $h = 45,\ 180,\ 900\,\mathrm{A/m}$, respectively. (b) With the direction fixed, the same positive magnitude deviation $\Delta b = 10^{-4}\,\mathrm{T}$ is applied at all three operating points, and Δh is computed directly from the same material table. The vertical axis is logarithmic, and the values beside the points show the actual material response. Blue-gray, orange, and dark blue correspond to I, II, and III, respectively.

For material m, magnetic flux density $\mathbf{B}$ and field strength $\mathbf{H}$ obey $\mathbf{H} = \mathcal{H}_m(\mathbf{B})$, with monotone magnitude relation $h = h_m(b)$, where $b = \|\mathbf{B}\|_2$ and $h = \|\mathbf{H}\|_2$. Fixing the field direction, consider a true magnitude $b^* > 0$ and prediction $b^* + \delta b$. Within a differentiable interval of h_m, Taylor expansion gives

$$\begin{aligned} \delta h &= h_m(b^* + \delta b) - h_m(b^*) = \nu_d\,\delta b + o(|\delta b|), \\ (\delta h)^2 &= \nu_d^2(\delta b)^2 + o((\delta b)^2), \qquad \nu_d = h_m'(b^*). \end{aligned} \tag{1}$$

Near the ground truth, squared error in physical H weights squared B-magnitude error by ν_d^2. This inverse constitutive slope is larger near saturation, so the same magnitude deviation incurs a larger penalty.

A vector field also has directional error. For the isotropic materials considered here, let $\mathbf{u} = \mathbf{B}^*/b^*$ and decompose the error as $\delta\mathbf{B} = \delta b_{\parallel}\mathbf{u} + \delta\mathbf{B}_{\perp}$, where $\mathbf{u}^{\top}\delta\mathbf{B}_{\perp} = 0$. Writing $\delta\mathbf{H} = \mathcal{H}_m(\mathbf{B}^* + \delta\mathbf{B}) - \mathcal{H}_m(\mathbf{B}^*)$, we obtain

$$\|\delta\mathbf{H}\|^2 = \nu_d^2(\delta b_{\parallel})^2 + \nu_s^2\|\delta\mathbf{B}_{\perp}\|^2 + o(\|\delta\mathbf{B}\|^2), \qquad \nu_s = \frac{h_m(b^*)}{b^*}. \tag{2}$$

The H coordinates thus weight magnitude and directional errors by material state. The weights derive from ν_d and ν_s, the eigenvalues of the differential reluctivity tensor $\mathbf{N} = \mathrm{D}_{\mathbf{B}}\mathcal{H}_m(\mathbf{B}^*)$ along and perpendicular to the field direction. This tensor enters the FEM Newton tangent matrix (Egger et al., 2026), connecting the prediction coordinates to the solver. Spatial coupling and damping also affect iteration, so acceleration still requires evaluation through complete solves.

We therefore predict H in magnetic materials and recover B through the inverse constitutive relation. We call this choice constitutive coordinate reparameterization. With $b_m = h_m^{-1}$,

$$\widehat{\mathbf{B}}^{(H)} = \mathcal{B}_m(\widehat{\mathbf{H}}) = \frac{b_m(\|\widehat{\mathbf{H}}\|)}{\|\widehat{\mathbf{H}}\|}\widehat{\mathbf{H}}. \tag{3}$$

The zero-field value is defined by continuity. Reversible encoding accommodates the range of H magnitudes across samples, so the material mapping and encoding jointly determine training error weights.

3.3 Range-Aware Encoding and Joint Multi-Head Learning

The wide range of H magnitudes can let strong-field errors dominate training and compress weak-field variations under a single linear scaling. We jointly predict field encodings and magnitude probabilities, recovering physical fields through a shared invertible function.

In Figure 3(a), material masks, current density, coordinates, external domain conditions, and multiscale excitation statistics feed a shared ConvNeXt-Tiny encoder and UPerNet decoder (Liu et al., 2022; Xiao et al., 2018). Spatial branches predict encoded B and H and log-relative secant permeability $\ell_\mu = \log(\mu/\mu_0)$. Magnitude branches predict high/low class probabilities $\mathbf{q}_B$ and $\mathbf{q}_H$, which condition the corresponding field decoding. The μ head contributes only auxiliary supervision to the shared features.

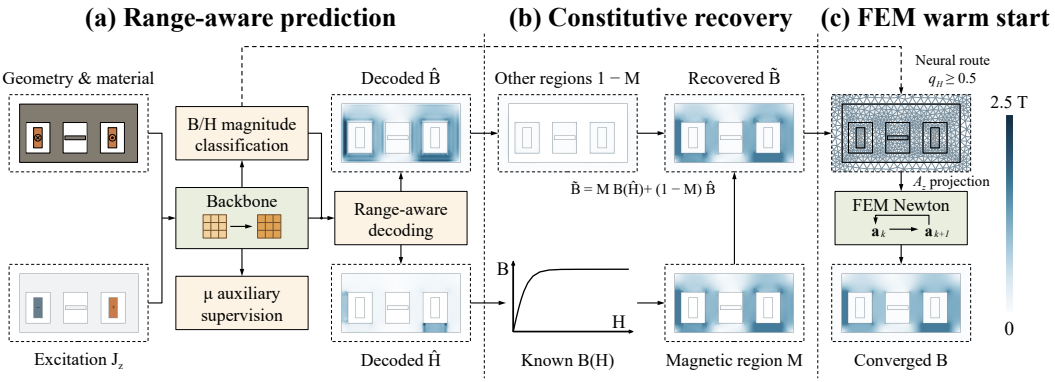


Figure 3: Zero-shot warm starts with CoRaWS on unseen topology families. (a) From geometry, materials, and excitation, a shared network predicts field encodings and magnitudes, then decodes B and H. The μ branch provides auxiliary supervision. (b) The upper branch retains predicted B in nonmagnetic regions. The lower branch recovers B from H in magnetic regions through the known $B(H)$ relation. Mask M combines both into a candidate field. (c) The upper dashed line carries the magnitude prediction that selects the initial guess. In the neural path shown, the candidate field is projected to an A_z initial guess, then corrected by FEM Newton iterations to meet the convergence criteria. Darker blue indicates larger B magnitudes in tesla (T) on the right-hand color bar. The H field uses a separate 0–3 $\mathrm{kA/m}$ scale.

Taking H as an example, let the spatial branch output $\mathbf{z}$ and write $r = \|\mathbf{z}\|$. Radial decoding is defined as

$$\widehat{\mathbf{H}} = s_H \sqrt{1+r^2} \exp\left(\gamma \frac{r^2}{1+r^2}\right) \mathbf{z}, \qquad \gamma = \sum_{c=0}^{1} \mathrm{sg}(q_c)\, a \tanh \beta_c, \quad a = \tfrac{1}{2}\log 2. \tag{4}$$

Here, s_H is a fixed physical scale, q_c is the c-th component of $\mathbf{q}_H$, and β_c is a learnable parameter. The decoded magnitude is approximately linear near zero and grows quadratically at large r, preserving weak-field variations while the inverse mapping compresses large magnitudes. The mapping preserves field direction and is strictly increasing in magnitude, with inverse E_γ used to encode labels. Both classes share this function and differ only in bounded γ. For fixed $\mathbf{z}$, any two class-dependent corrections have a magnitude ratio of at most 2, bounding the effect of classification on recovered magnitude rather than total prediction error.

Magnitude labels use each sample's root-mean-square field magnitude, with thresholds and normalization scales fitted only on source-family training data. Predicted probabilities determine γ during both training and inference. The operator sg blocks field-loss gradients through the probability mixture into the classifier. Cross-entropy supervises the classifier, while field regression learns β_c. The B branch uses the same encoding form with its own scale and probabilities.

Training combines the three spatial supervision terms with magnitude classification:

$$L_{\mathrm{dir}} = L_B + L_H + L_\mu + 0.1\big[\mathrm{CE}(\mathbf{q}_B, c_B^*) + \mathrm{CE}(\mathbf{q}_H, c_H^*)\big]. \tag{5}$$

The terms L_B, L_H, and L_μ use squared errors to compare B encodings in the valid region, normalized H encodings in regions with consistent magnetic-material labels, and ℓ_μ in that same region, respectively. The labels $\mathbf{H}^* = \mathcal{H}_m(\mathbf{B}^*)$ and μ are both obtained from the original B labels using the same material table, without additional field solves.

3.4 Zero-Shot Initialization and Selective Warm Starts

Field recovery and projection add overhead, so fewer iterations need not reduce total cost. The magnitude prediction from the same forward pass selects the initialization path, as shown by the upper dashed line in Figure 3. For neural initialization, the geometric magnetic-material mask M combines constitutive recovery from Equation (3) with the B-head prediction elsewhere, as shown in Figure 3(b):

$$\widetilde{\mathbf{B}} = M\, \mathcal{B}_m(\widehat{\mathbf{H}}) + (1-M)\widehat{\mathbf{B}}. \tag{6}$$

FEM updates magnetic vector potential coefficients, requiring the projection in Figure 3(c). We fit $\mathcal{C}A_z = (\partial_y A_z, -\partial_x A_z)^\top$ to $\widetilde{\mathbf{B}}$ in the interior domain, harmonically extend the potential into current-free exterior air, and fix its additive constant using the outer boundary and known loads. This gives finite element coefficients $\Pi(\widetilde{\mathbf{B}})$ for Newton iteration.

Let q_H denote the probability of the high-H class in $\mathbf{q}_H$. The final initial guess is

$$\mathbf{a}_0 = \begin{cases} \mathbf{0}, & q_H < 0.5, \\ \Pi(\widetilde{\mathbf{B}}), & q_H \geq 0.5. \end{cases} \tag{7}$$

All target samples use this threshold. Both paths include the cost of one network forward pass, while the zero-initialization path skips candidate-field combination and projection. The probability q_H reflects field magnitude, not local saturation fraction. Ablations examine its roles in learning and path selection.

The full data construction procedure and splits are given in Appendix B, and the constitutive and encoding derivations and initialization details are given in Appendix C.

4 Experiments

4.1 Experimental Setup and Evaluation Protocol

For each of three splits, we train on 400 samples from each of 30 source families and randomly select 20 held-out samples from each of the other 24 target families, without filtering by magnitude or solution difficulty, yielding 1,440 evaluation pairs and 1,080 distinct samples.

A pair denotes one evaluation sample in one split, so the same sample may occur in different splits. The neural models share a ConvNeXt-Tiny encoder and a UPerNet-based decoder (Liu et al., 2022; Xiao et al., 2018), with the same parameter initialization and a training budget of 120 epochs.

Following the use of neural predictions to initialize Newton iteration (Taghikhani et al., 2026), Direct-B projects its predicted magnetic field to an initial guess, whereas Direct-A_z learns a potential proxy obtained by projecting Maxwell B and then transfers it to FEM degrees of freedom. POD-ridge-A_z predicts the same proxy using POD and ridge regression (Kramer & Qian, 2026). These learned methods use only source-family labels, while nearest-neighbor warm starts retrieve a solution by parameter distance from five support samples in the target family (Sambharya et al., 2024). C2F adaptively refines and solves a coarse mesh for each sample, then transfers its magnetic vector potential to the common fine mesh for Newton iteration (Bringmann et al., 2025; Adler et al., 2010).

All methods use the same material table and fine triangular mesh for a given sample, with quadratic Lagrange elements (P_2) for A_z and quadrature order 10. The minimum target size at corners is $0.0025\,\mathrm{mm}$, and the bulk size starts at $0.25\,\mathrm{mm}$ and is adjusted according to geometry and element count. Each solve uses one CPU thread and requires a relative residual below 10^{-8} and a final-field difference below 10^{-6} relative to the FEM reference solution on the same mesh. Initial-guess error is also measured against this FEM solution, while labels for training across families come from Maxwell. We apply these acceptance criteria and compare total online cost, including initialization overhead (McGreivy & Hakim, 2024):

$$T_{\mathrm{online}} = T_{\mathrm{setup}} + T_{\mathrm{input}} + T_{\mathrm{net}} + T_{\mathrm{init}} + T_{\mathrm{solve}} + T_{\mathrm{out}}. \tag{8}$$

The terms correspond, in order, to mesh and system preparation, input construction, forward inference and data transfer, initial-guess construction, numerical iteration, and result output. For C2F, this includes mesh generation, all intermediate solves, error estimation, refinement, and transfer.

4.2 Comparison with Other Methods

As shown in Table 1, CoRaWS satisfies the acceptance criteria on all 1,440 evaluation pairs and reduces cumulative online time by 8.41% and total Newton updates by 11.79% relative to zero initialization. These savings include network inference and initial-guess construction, and require neither target-family labels nor fine-tuning. CoRaWS achieves the lowest cumulative online cost among the evaluated learned and retrieval-based methods. By comparison, both Direct-B and Direct-A_z take longer than zero initialization, showing that replacing a zero initial guess with a learned prediction does not automatically produce a net speedup.

Table 1: Solver results on all 1,440 pairs. *Init. error* denotes median initial error: the median relative L^2 error (%) of B entering the fine-mesh solve against the FEM reference. *Time* is the cumulative online time in hours. *Speedup* is the ratio of cumulative time for zero initialization to that for each method. *Extra setup* denotes preparation outside per-query online timing, excluding source-family training and common FEM settings.

Method	Extra setup	Init. error (%)	Time (h) ↓	Speedup ↑
Zero start	None	100.00	19.4688	1.0000
Direct-B	None	56.71	20.6191	0.9442
Direct-A_z	None	103.26	22.2866	0.8736
POD-ridge-A_z	None	7,849.93	39.8577	0.4885
Nearest neighbor	Support solves	77.06	24.7077	0.7880
C2F	None	0.94	18.6659	1.0430
CoRaWS	None	100.00	17.8321	1.0918

Initial-field error alone does not explain this ordering. Direct-B reduces median initial error from 100% to 56.71%, yet increases cumulative online time from 19.4688 h to 20.6191 h. For CoRaWS, the median initial error remains 100% because routing assigns zero initialization to 1,121 pairs. This value describes the initial fields supplied to FEM after routing, rather than the accuracy of the network predictions alone. The comparison therefore needs to account for both the quality of an initial field and the cost of constructing it.

C2F makes this distinction particularly clear. It achieves the lowest median initial error, 0.94%, but requires 18.6659 h once mesh refinement, intermediate solves, and transfer are included. CoRaWS requires 17.8321 h, a reduction of 4.47%. Thus, a more accurate initial field need not lead to a lower total solution cost. The following ablations examine how constitutive recovery and routing contribute to this balance. Results for each topology-family split, including the slight slowdown in Fold 1, are reported in Appendix B.3.

4.3 Ablation Studies

Our ablations address two questions: whether the prediction representation actually reduces nonlinear iterations, and whether iteration savings cover the cost of constructing the initial guess. We first conduct a controlled experiment supervised by FEM solutions from a single topology family, keeping the three-head model weights fixed and replacing direct use of the B head with prediction of H followed by constitutive recovery of B. The mean Newton step count over 20 cases decreases from 18.35 to 15.30, providing direct evidence for using H coordinates and material-based recovery. As shown in Table 2, the experiments across topology families first disable routing to compare output representations, magnitude encodings, and supervision objectives, then fix the CoRaWS weights to examine the effect of routing.

Table 2: Zero-shot ablations on the same 1,440 pairs, with error and timing metrics as in Table 1. *Mean iters.* denotes the mean number of fine-mesh Newton updates. Except for Direct-B, all variants supervise $B/H/\mu$. These three-head variants recover B from H in magnetic materials unless marked B head. Linear H uses fixed linear scaling. Shared H encoding uses Equation (4) with a shared γ and no H classification supervision. + class loss adds this supervision, and CoRaWS further conditions the encoding on class probabilities. Cross supervision matches $\widehat{\mu}\widehat{\mathbf{H}}$ and $\widehat{\mathbf{B}}/\widehat{\mu}$ to their labels. The final four rows share CoRaWS weights within each split. Routing Off always uses a neural initial guess. Routing On uses identical q_H decisions for both recovery paths according to Equation (7).

Variant	Routing	Init. error (%) ↓	Mean iters. ↓	Time (h) ↓
Direct-B	Off	56.71	5.00	20.6191
Linear H	Off	928.67	5.96	23.1499
Shared H encoding	Off	66.87	4.74	20.5373
Shared H encoding + class loss	Off	63.69	4.69	20.4871
CoRaWS + cross supervision	Off	66.03	4.78	20.0340
CoRaWS, B head	Off	66.88	5.01	21.3054
CoRaWS	Off	66.01	4.70	19.8110
CoRaWS, B head	On	100.00	5.01	19.2406
CoRaWS	On	100.00	4.69	17.8321

To test whether the recovery advantage extends across topology families, we compare the two outputs of the same CoRaWS model under identical routing. Both variants satisfy final acceptance on all 1,440 pairs, but constitutive recovery reduces total Newton updates from 7,219 to 6,759, a reduction of 6.37%. Because the weights and routing decisions are unchanged, this comparison supports a contribution from constitutive recovery beyond sample selection alone.

With routing disabled, CoRaWS reduces the mean step count from 5.00 for Direct-B to 4.70, even though median initial error increases from 56.71% to 66.01%. Magnitude encoding also affects the effectiveness of the H representation. Fixed linear scaling yields an initial error of 928.67% and requires 5.96 steps on average, both worse than Direct-B. With the same three-head supervision and constitutive recovery, replacing linear scaling with shared nonlinear encoding reduces the error to 66.87% and the step count to 4.74. This shows that magnitude differences across families affect the performance of the H representation, supporting the use of nonlinear magnitude encoding in addition to constitutive coordinates.

Adding classification supervision to the shared encoding reduces error from 66.87% to 63.69% and the step count from 4.74 to 4.69, showing that the auxiliary magnitude task can provide a modest improvement. Further adjusting the encoding by class gives CoRaWS a mean step count of 4.70, with no additional reduction. With the same representation and recovery method, adding cross supervision leaves initial error nearly unchanged and instead increases the mean step count from 4.70 to 4.78.

The final model therefore retains direct supervision of the three heads and recovers the magnetic field through the known material relation, without a cross-supervision term.

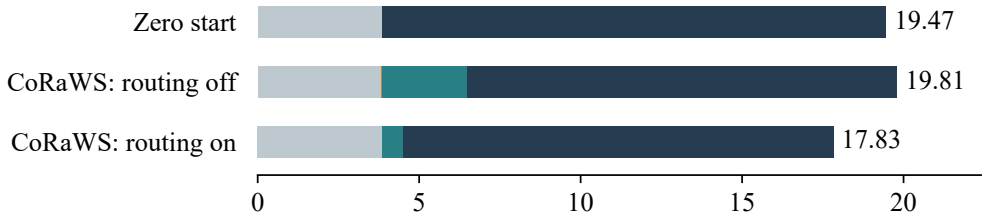


Figure 4: Cumulative online cost with fixed weights (1,440 pairs, in hours). Routing mainly reduces projection overhead. Light gray, orange, teal, and dark blue denote shared preparation and export, input construction and inference/data transfer, projection preparation and projection, and FEM iterations, respectively.

The iteration savings above must still cover the cost of constructing the initial guess. CoRaWS without routing takes 19.811 h, more than the 19.469 h required by zero initialization. The classification branch serves another purpose here: the high-H probability distinguishes high- and low-iteration groups with an AUROC of 0.986, allowing CoRaWS to decide whether to construct a neural initial guess. As shown in Figure 4, enabling routing with fixed weights changes the mean step count only from 4.70 to 4.69, but reduces projection-related time from 2.646 h to 0.621 h and cumulative online time to 17.832 h. Neural initialization reduces nonlinear iterations, while routing avoids construction overhead for samples that are inexpensive to solve from zero. In this implementation, combining the two converts iteration savings into a reduction in total online cost.

Training and evaluation settings are given in Appendix B, baseline constructions in Appendix C.6, and controlled single-family experiments in Appendix D.

5 Conclusion

We introduce EMTopoGen and CoRaWS to accelerate nonlinear magnetostatic solves without target-family labels or fine-tuning. Our analysis relates prediction coordinates and encoding to local error weights, and experiments assess complete solves under matched meshes and convergence criteria. Across 1,440 evaluation pairs in three splits, CoRaWS reduces cumulative online time by 8.41% relative to zero initialization, including inference and initial-guess construction. These results highlight the need to consider physical representation, initialization, and subsequent iteration together. Although evaluation is limited to magnetostatics, separating primitive shapes, assembly relations, and continuous parameters may support compositional benchmarks in other engineering domains. Likewise, choosing coordinates through known constitutive relations can weight prediction errors by physical response. Future work will examine physical coordinates and their relation to iteration in nonlinear heat transfer and solid mechanics, and test whether these principles yield consistent online gains across a broader range of materials and structures.

AI USE STATEMENT

In this study, we used generative AI tools to assist with literature searches. These tools were also used for manuscript translation and language editing. Physical field labels were computed using Maxwell and the finite element solver presented in this paper, rather than generated by language models. Checks of AI-assisted content included code testing, numerical consistency checks, comparisons with experimental records, and verification of references against primary sources. The authors led the research, made the final methodological and editorial decisions, and take responsibility for the methods, results, and claims presented in this paper.

REFERENCES

J. H. Adler, T. A. Manteuffel, S. F. McCormick, J. W. Ruge, and G. D. Sanders. Nested iteration and first-order system least squares for incompressible, resistive magnetohydrodynamics. *SIAM Journal on Scientific Computing*, 32(3):1506–1526, 2010. doi: 10.1137/090766905. URL https://jadler.math.tufts.edu/publications/files/2010_MHD_NI_SISC.pdf.

Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015. doi: 10.1137/130932715. URL https://epubs.siam.org/doi/10.1137/130932715.

Johannes Brandstetter, Max Welling, and Daniel E. Worrall. Lie point symmetry data augmentation for neural PDE solvers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2241–2256, 2022. URL https://proceedings.mlr.press/v162/brandstetter22a.html.

Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3 edition, 2008. doi: 10.1007/978-0-387-75934-0. URL https://link.springer.com/book/10.1007/978-0-387-75934-0.

Philipp Bringmann, Michael Feischl, Ani Miraci, Dirk Praetorius, and Julian Streitberger. On full linear convergence and optimal complexity of adaptive FEM with inexact solver. *Computers & Mathematics with Applications*, 180:102–129, 2025. doi: 10.1016/j.camwa.2024.12.013. URL https://doi.org/10.1016/j.camwa.2024.12.013.

Herbert Egger, Felix Engertsberger, and Bogdan Radu. On the convergence of higher-order finite element methods for nonlinear magnetostatics. *SIAM Journal on Numerical Analysis*, 64(1):55–75, 2026. doi: 10.1137/24M168814X. URL https://doi.org/10.1137/24M168814X.

Mohammad Sadegh Eshaghi, Cosmin Anitescu, Navid Valizadeh, Yizheng Wang, Xiaoying Zhuang, and Timon Rabczuk. NOWS: Neural operator warm starts for accelerating iterative solvers. *arXiv preprint arXiv:2511.02481*, 2026. URL https://arxiv.org/abs/2511.02481. Version 4.

Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. doi: 10.1002/nme.2579. URL https://doi.org/10.1002/nme.2579.

Tom Gustafsson and G. D. McBain. scikit-fem: A python package for finite element assembly. *Journal of Open Source Software*, 5(52):2369, 2020. doi: 10.21105/joss.02369. URL https://joss.theoj.org/papers/10.21105/joss.02369.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2): 217–288, 2011. doi: 10.1137/090771806. URL https://arxiv.org/abs/0909.4061.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634. URL https://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634.

Benjamin Holzschuh, Qiang Liu, Georg Kohl, and Nils Thuerey. PDE-Transformer: Efficient and versatile transformers for physics simulations. In *Proceedings of the International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 23562–23602, 2025. URL https://proceedings.mlr.press/v267/holzschuh25a.html.

Jianing Huang, Kaixuan Zhang, Youjia Wu, and Ze Cheng. Operator learning with domain decomposition for geometry generalization in PDE solving. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/710445227fa8c1b6a9ceada902dd4741-Abstract-Conference.html.

Boris Kramer and Elizabeth Qian. Reduced-order modeling for engineering systems: survey and opportunities for digital twins. *Structural and Multidisciplinary Optimization*, 69:141, 2026. doi: 10.1007/s00158-026-04310-0. URL https://link.springer.com/article/10.1007/s00158-026-04310-0.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.08895.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS Journal of Data Science*, 2024. doi: 10.1145/3648506. URL https://authors.library.caltech.edu/records/06m4r-8jf10.

Qibang Liu, Weiheng Zhong, Hadi Meidani, Diab Abueidda, Seid Koric, and Philippe Geubelle. Geometry-informed neural operator transformer for partial differential equations on arbitrary geometries. *Computer Methods in Applied Mechanics and Engineering*, 451:118668, 2026a. doi: 10.1016/j.cma.2025.118668. URL https://doi.org/10.1016/j.cma.2025.118668.

Shouyi Liu, Xiaokang Yang, and Yuntian Chen. Riesz neural operator for solving partial differential equations. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Liu_A_ConvNet_for_the_2020s_CVPR_2022_paper.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Huakun Luo, Haixu Wu, Hang Zhou, Lanxiang Xing, Yichen Di, Jianmin Wang, and Mingsheng Long. Transolver++: An accurate neural solver for PDEs on million-scale geometries. In *Proceedings of the International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41432–41449, 2025. URL https://proceedings.mlr.press/v267/luo25o.html.

Siying (Sydney) Ma, Mehrdad Momeni Zadeh, Mauricio Soroco, Wuyang Chen, Jiguo Cao, and Vijay Ganesh. Learning data-efficient and generalizable neural operators via fundamental physics knowledge. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/1b6554ab420ad26328d73e3387613f4e-Abstract-Conference.html.

Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence*, 6: 1256–1269, 2024. doi: 10.1038/s42256-024-00897-5. URL https://www.nature.com/articles/s42256-024-00897-5.

Marcel Nonnenmacher and Nicolas Durrande. AI accelerated FEM solvers: Improving convergence with better initialisation. Monumo technical article, 2026. URL https://monumo.ai/latest/ai-accelerated-fem-solvers-improving-convergence-with-better-initialisation. Published June 24, 2026; accessed September 21, 2026.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The well: A large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/4f9a5acd91ac76569f2fe291b1f4772b-Abstract-Datasets_and_Benchmarks_Track.html.

Fabian J. Roth, Dominik K. Klein, Maximilian Kannapinn, Jan Peters, and Oliver Weeger. Stable port-Hamiltonian neural networks. *arXiv preprint arXiv:2502.02480*, 2025. URL https://arxiv.org/abs/2502.02480. Version 2.

Rajiv Sambharya, Georgina Hall, Brandon Amos, and Bartolomeo Stellato. Learning to warm-start fixed-point optimization algorithms. *Journal of Machine Learning Research*, 25(166):1–46, 2024. URL https://www.jmlr.org/papers/v25/23-1174.html.

Zakhar Shumaylov, Peter Zaika, James Rowbottom, Ferdia Sherry, Melanie Weber, and Carola-Bibiane Schönlieb. Lie algebra canonicalization: Equivariant neural operators under arbitrary lie groups. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/056521a35eacd9d2127b66a7d3c499c5-Abstract-Conference.html.

Kianoosh Taghikhani, Yusuke Yamazaki, Jerry Paul Varghese, Markus Apel, Reza Najian Asl, and Shahed Rezaei. Neural-initialized Newton: Accelerating nonlinear finite elements via operator learning. *International Journal for Numerical Methods in Engineering*, 127(14):e70380, 2026. doi: 10.1002/nme.70380. URL https://onlinelibrary.wiley.com/doi/10.1002/nme.70380.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2022. URL https://arxiv.org/abs/2210.07182.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2. URL https://scipy.org/citing-scipy/.

Moritz von Tresckow, Ion Gabriel Ion, and Dimitrios Loukrezis. Multi-patch neural solver with isogeometric mappings for partial differential equations on computer-aided design domains. *Engineering with Computers*, 42(4):116, 2026. doi: 10.1007/s00366-026-02351-z. URL https://doi.org/10.1007/s00366-026-02351-z.

Sifan Wang, Jacob Seidman, Shyam Sankaran, Hanwen Wang, George Pappas, and Paris Perdikaris. CViT: Continuous vision transformer for operator learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3bf4b55960aaa23553cd2a6bdc6e1b57-Abstract-Conference.html.

Yizheng Wang, Zhongkai Hao, Mohammad Sadegh Eshaghi, Cosmin Anitescu, Xiaoying Zhuang, Timon Rabczuk, and Yinghua Liu. Pretrain finite element method: A pretraining and warm-start framework for PDEs via physics-informed neural operators. *Journal of the Mechanics and Physics of Solids*, 214:106682, 2026. doi: 10.1016/j.jmps.2026.106682. URL https://doi.org/10.1016/j.jmps.2026.106682.

Shizheng Wen, Arsh Kumbhat, Levi Lingsch, Sepehr Mousavi, Yizhou Zhao, Praveen Chandrashekar, and Siddhartha Mishra. Geometry aware operator transformer as an efficient and accurate neural surrogate for PDEs on arbitrary domains. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=HXFvNkNt0n.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision*, 2018. URL https://www.ecva.net/papers/eccv_2018/papers_ECCV/html/Tete_Xiao_Unified_Perceptual_Parsing_ECCV_2018_paper.php.

Zhizhou Zhang, Youjia Wu, Kaixuan Zhang, and Yanjia Wang. From cheap geometry to expensive physics: A physics-agnostic pretraining framework for neural operators. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/df656d6ed77b565e8dcdfbf568aead0a-Abstract-Conference.html.

A Magnetostatic Formulation and Warm-Start Task

A.1 Nonlinear Magnetostatics

Consider a two-dimensional computational domain $\Omega \subset \mathbb{R}^2$ containing a device and the surrounding air. The material distribution $m(\mathbf{x})$ and out-of-plane current density $J_z(\mathbf{x})$ are known, and the unknown magnetic flux density $\mathbf{B}$ and magnetic field strength $\mathbf{H}$ lie in the plane and satisfy

$$\partial_x H_y - \partial_y H_x = J_z, \qquad \nabla \cdot \mathbf{B} = 0. \tag{9}$$

We use a single-valued, isotropic material relation without hysteresis. Writing $b = \|\mathbf{B}\|_2$ and $h = \|\mathbf{H}\|_2$, we describe the material response by a monotone curve $h_m(b)$:

$$\mathbf{H} = \mathcal{H}_m(\mathbf{B}) = \nu_m(b)\mathbf{B}, \qquad \nu_m(b) = \frac{h_m(b)}{b}, \qquad \mu_m(b) = \frac{1}{\nu_m(b)}. \tag{10}$$

Here, μ_m and ν_m are the secant permeability and secant reluctivity, respectively, with their zero-field values defined by continuity. Air and linear materials have constant permeability, while soft magnetic materials use piecewise linear interpolation of a given B–H table. As magnetization approaches saturation, the incremental material response changes, making the field equations nonlinear. Introducing the out-of-plane magnetic vector potential A_z and defining the two-dimensional curl operator $\mathcal{C}v = (\partial_y v, -\partial_x v)^\top$, the representation $\mathbf{B} = \mathcal{C}A_z$ automatically satisfies the divergence constraint. Two-dimensional magnetostatics can thus be described by the scalar unknown A_z (Egger et al., 2026), and we impose $A_z = 0$ on the outer boundary of the truncated domain.

A.2 Finite Element Discretization and Damped Newton Solves

We use a conforming finite element discretization on a triangular mesh $\mathcal{T}_h$ (Brenner & Scott, 2008). After eliminating boundary degrees of freedom, we express the magnetic vector potential in basis functions ϕ_i as $A_{z,h} = \sum_i a_i \phi_i$. Let $\mathbf{C}_q\mathbf{a}$ denote the magnetic field at quadrature point q, w_q the quadrature weight including the geometric mapping, and $\mathbf{f}$ the current load vector. The discrete residual and its tangent matrix are

$$\begin{aligned} \mathbf{r}(\mathbf{a}) &= \sum_q w_q \mathbf{C}_q^\top \mathcal{H}_{m_q}(\mathbf{C}_q\mathbf{a}) - \mathbf{f}, \\ \mathbf{K}(\mathbf{a}) &= \sum_q w_q \mathbf{C}_q^\top \mathbf{N}_q \mathbf{C}_q, \qquad \mathbf{N}_q = \mathrm{D}_{\mathbf{B}} \mathcal{H}_{m_q}(\mathbf{C}_q\mathbf{a}). \end{aligned} \tag{11}$$

Here, $\mathbf{N}_q$ is the differential reluctivity tensor, which describes the change in $\mathbf{H}$ induced by a local change in $\mathbf{B}$ and differs from the secant reluctivity ν_m. We use one-sided derivatives at the knots of the material table. Given an initial guess $\mathbf{a}_0$ satisfying the boundary conditions, the damped Newton update is

$$\mathbf{K}(\mathbf{a}_k)\Delta\mathbf{a}_k = -\mathbf{r}(\mathbf{a}_k), \qquad \mathbf{a}_{k+1} = \mathbf{a}_k + \alpha_k \Delta\mathbf{a}_k, \quad 0 < \alpha_k \le 1. \tag{12}$$

A backtracking line search selects the step size based on a decrease in discrete magnetic energy or residual, and iteration stops when $\|\mathbf{r}(\mathbf{a}_k)\|_2 / \|\mathbf{f}\|_2 < \varepsilon_r$.

A.3 Zero-Shot Warm Starts across Topology Families

A topology family is defined by its component composition and assembly-relation template, while continuous changes in dimensions and excitations within a family are parameter variations. Let the source and target family sets be $\mathcal{F}_{\mathrm{src}}$ and $\mathcal{F}_{\mathrm{tgt}}$, respectively, with $\mathcal{F}_{\mathrm{src}} \cap \mathcal{F}_{\mathrm{tgt}} = \varnothing$. The model and data-driven preprocessing are fitted only to source-family training data. For a target family, the initialization mapping $\mathbf{a}_0 = \mathcal{I}_\theta(\xi, \mathcal{T}_h)$ generates an initial guess from the known geometry, materials, excitations, and boundary conditions ξ, after which the original FEM solver continues the solve. Zero-shot means that this process neither accesses target-family field labels nor fine-tunes on target families.

All initial guesses use the same mesh, materials, iteration rules, and convergence thresholds, and the converged field must have a relative difference below ε_B from the reference solution on the same mesh. Online cost is defined in Equation (8). The initial guess itself need not satisfy the convergence criteria, as the objective is to reduce the total cost of reaching those criteria with the original FEM solver.

B Dataset Construction and Experimental Protocol

B.1 Topology Families and Holdout Splits

EMTopoBench-54 organizes topology families by engineering function into nine groups: composite multipole magnets, flux concentration and diversion, defect-scanning magnetizers, multi-window bypass magnetic circuits, open multi-coil magnetic circuits, planar segmented flux-diversion circuits, shaped poles, shielding and nested structures, and variable-air-gap devices. Each group contains six distinct assembly templates, each defining a topology family. Continuous parameters vary dimensions and excitations within a template, while component combinations and spatial relations change across templates. Geometry is defined separately from the assignment of materials and excitations, and material identity is specified by physical regions rather than inferred from geometric identifiers. Each family retains 500 samples under the same numerical acceptance criteria, with additional sampling replacing candidates that fail these criteria.

Each family is divided into 400 training samples and 100 held-out samples. Each topology-family split holds out three entire engineering groups and selects five families from each of the remaining six groups as source families, giving 30 source families in total. The one remaining family from each of these six groups and the 18 families from the fully held-out groups form the 24 target families. Twenty held-out samples per source family are used to monitor training. For each target family, 20 evaluation samples are randomly drawn from its held-out pool. These target-family evaluation samples are not used to train the model for the corresponding split. However, F32 was used for method development, and five of its evaluation samples overlap with the single-family development set. We therefore also report results excluding F32 in Appendix D.5. The same evaluation samples are used for a given target family across splits, and all neural models share the source-family training data, shared-parameter initialization, and training budget. The three splits yield 1,440 evaluation pairs corresponding to 1,080 distinct physical samples.

The target families include unseen assembly templates within seen engineering groups as well as entirely held-out engineering groups. Of all evaluation pairs, 120 come from entirely held-out engineering groups and include circles or octagons not seen in the source families, while the other 1,320 pairs use primitive shape types seen during training. The evaluation thus includes both unseen structural combinations and unseen primitive shapes, while dimensional variation within a family is not counted as topological variation.

B.2 Training and Finite Element Solver Settings

Table 3 lists the shared settings.

Table 3: Training and solver settings for experiments across topology families. All neural prediction models use the same training budget and are evaluated using their parameters at epoch 120.

Item	Setting
Dataset	9 engineering groups, 54 families, 500 samples per family
Source split	3 folds, seed 17, 30 source families per fold, 400 training samples per source family
Target queries	24 target families per fold, 20 fixed queries per family
Evaluation set	1,440 pairs, 1,080 distinct samples, drawn from target-family held-out pools
Training budget	120 epochs, batch size 16, AdamW (Loshchilov & Hutter, 2019)
Optimizer	Learning rate 3×10^{-4}, weight decay 10^{-4}
Training labels	Maxwell B, with H and μ obtained from the known material curve
FEM discretization	Quadratic Lagrange triangles (P_2), quadrature order 10
Solver budget	One CPU thread per solve, at most 80 Newton steps
Acceptance	Relative residual $< 10^{-8}$, final-field relative error $< 10^{-6}$ against the same-mesh FEM reference

Source families are supervised by Maxwell B fields, from which H and μ are obtained using the same material relation. For each solve, Gmsh generates the triangular mesh (Geuzaine & Remacle, 2009), scikit-fem performs finite element assembly (Gustafsson & McBain, 2020), and SciPy solves

the Newton linear systems (Virtanen et al., 2020). Different initialization methods for the same sample share the mesh, material relation, and acceptance criteria.

Our evaluation measures online deployment of a pretrained model. Source-family data generation and pretraining are offline preparation costs that can be amortized over subsequent solves. For each source-family split, the same trained model is reused across unseen target families without acquiring target-family labels or performing fine-tuning.

B.3 Results by Topology-Family Split

Table 4 reports the results separately for each topology-family split. Each split contains 480 evaluation pairs. The high-iteration group consists of pairs requiring at least 10 Newton iterations from zero initialization. Both zero initialization and CoRaWS satisfy the final acceptance criteria on all pairs. The CoRaWS results use magnitude-based routing. Its online time combines the recorded costs of the selected initialization and solver stages, including input construction and network inference.

Table 4: Results by topology-family split. Times are cumulative online costs, and the last two columns report mean Newton iterations. Time saving is computed as $100(1 - T_{\mathrm{CoRaWS}}/T_0)$, where T_0 is the zero-initialization cost. Negative values indicate a slowdown.

Fold	Pairs	Zero (h)	CoRaWS (h)	Saving (%)	Zero iters.	CoRaWS iters.
All evaluation pairs						
0	480	4.2335	3.8089	10.03	3.194	2.729
1	480	7.1247	7.1813	−0.79	5.444	5.340
2	480	8.1106	6.8418	15.64	7.325	6.013
Pooled	1,440	19.4688	17.8321	8.41	5.321	4.694
High-iteration group						
0	40	1.6967	1.2666	25.35	21.150	15.575
1	100	4.3575	3.6934	15.24	21.000	17.200
2	141	5.9553	4.6821	21.38	21.234	16.766
Pooled	281	12.0095	9.6421	19.71	21.139	16.751

CoRaWS reduces mean Newton iterations in all three topology-family splits. For the high-iteration group, it reduces both iteration counts and online time in every split. The time reductions are 25.35%, 15.24%, and 21.38% for Folds 0, 1, and 2, respectively. These results show that the gains on computationally demanding samples are observed across all three evaluated splits.

Across all evaluation pairs, Folds 0 and 2 achieve time savings of 10.03% and 15.64%, respectively. In Fold 1, the high-iteration group saves 39.84 minutes, while the remaining samples add 43.24 minutes, resulting in a small net increase of 0.79%. This difference reflects the balance between savings on expensive solves and additional costs on other samples. Pooling the three splits, CoRaWS reduces cumulative online time by 8.41% and total Newton iterations by 11.79%.

B.4 Visualization of Topologies and Splits

Figures 5–7 show all 54 topology families, complementing the selected examples in Figure 1. The first generated instance of each family is used, without selection based on field error or solver gains. Each instance preserves its gaps, containment relations, and aspect ratio and is scaled individually for structural comparison. The three markers beside each family indicate its source or target status in folds 0, 1, and 2, respectively.

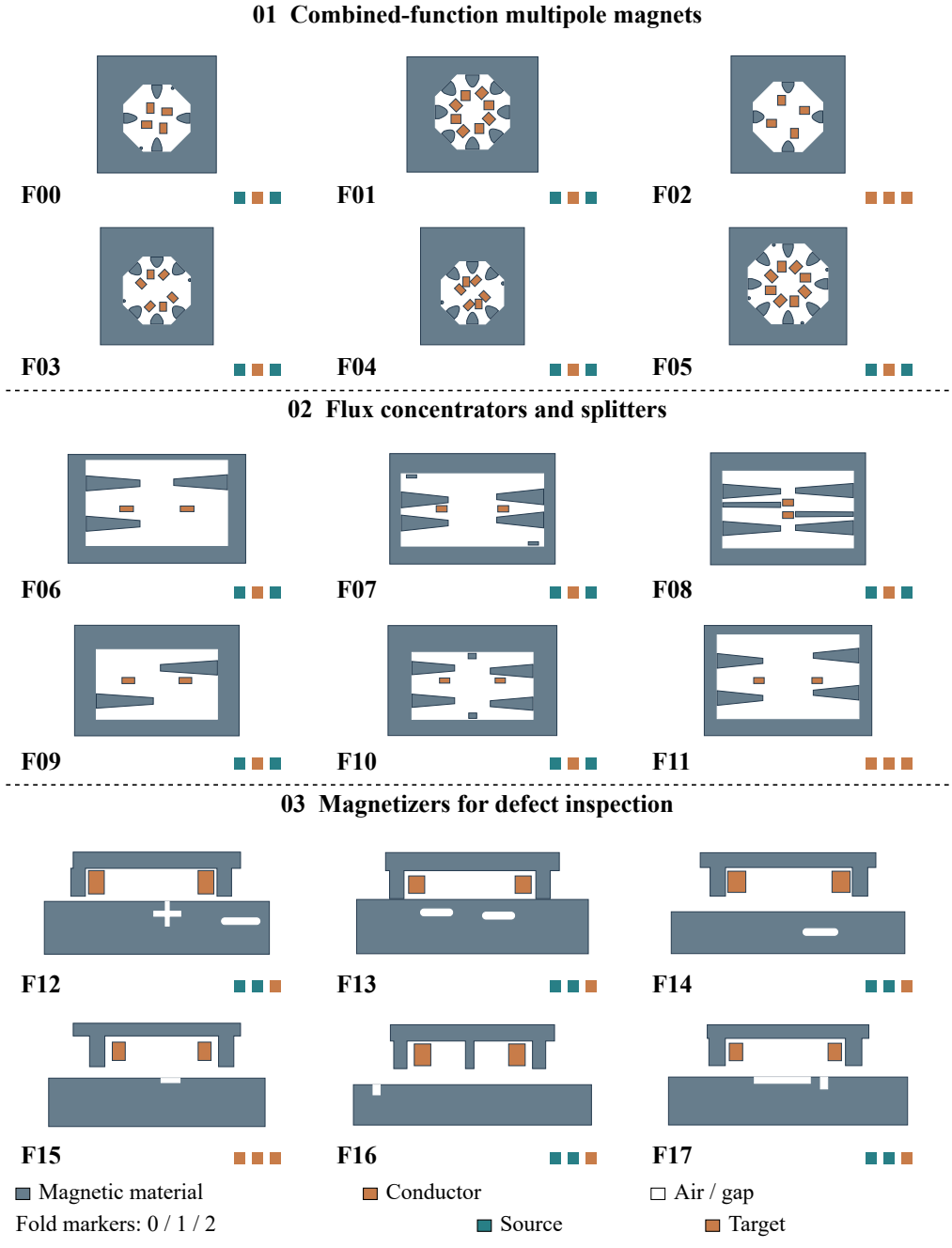


Figure 5: Topology families F00–F17 in EMTopoBench-54, covering composite multipole magnets, flux concentration and diversion, and defect-scanning magnetizers. Blue-gray denotes magnetic material, orange denotes conductors, and white denotes air or gaps. The three color blocks beside each family indicate its status in folds 0, 1, and 2, respectively, with teal for source families and orange for target families. Each instance is scaled individually while preserving its aspect ratio.

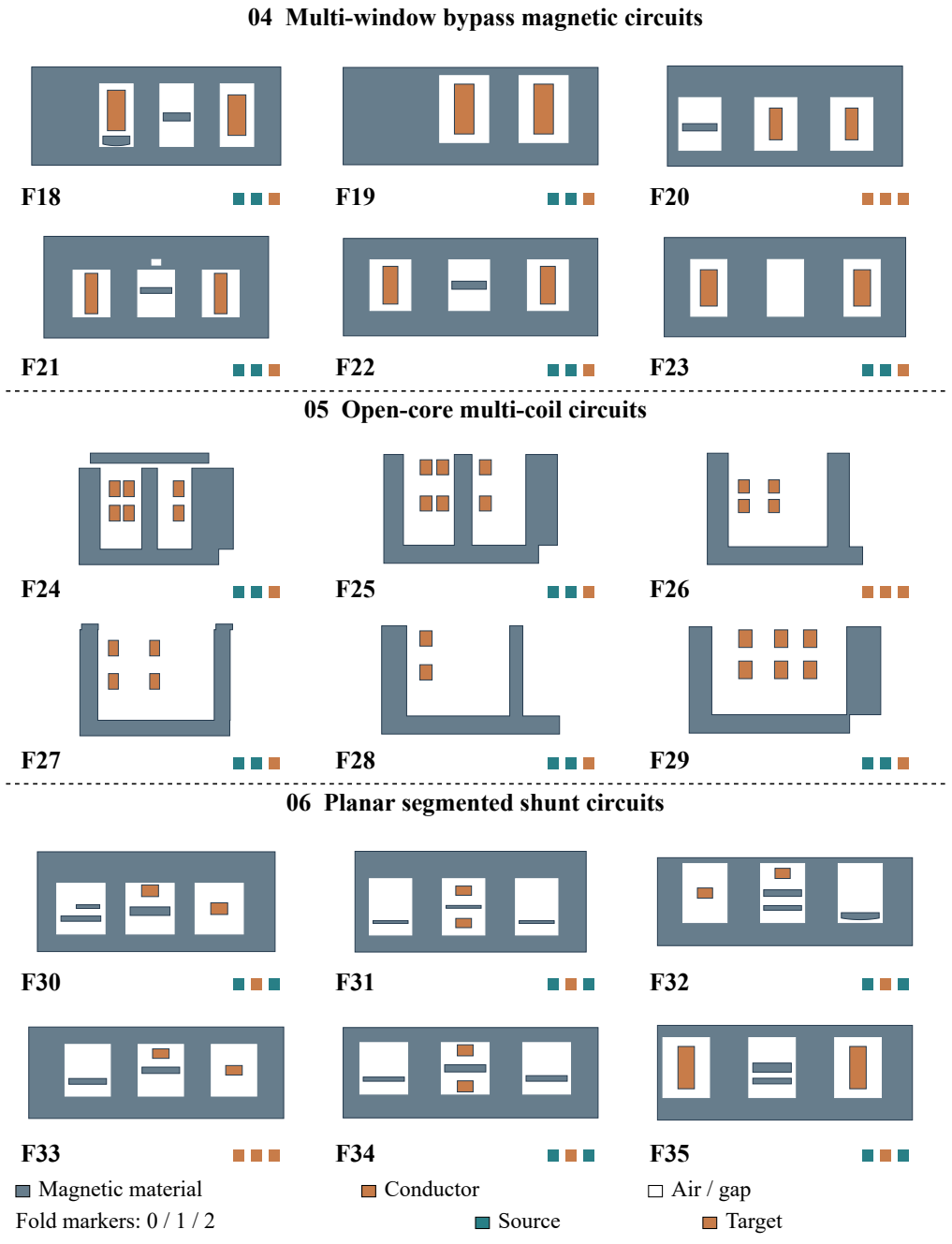


Figure 6: Topology families F18–F35 in EMTopoBench-54, covering multi-window bypass, open multi-coil, and planar segmented flux-diversion magnetic circuits. Material colors, split markers, and scaling conventions are the same as in Figure 5.

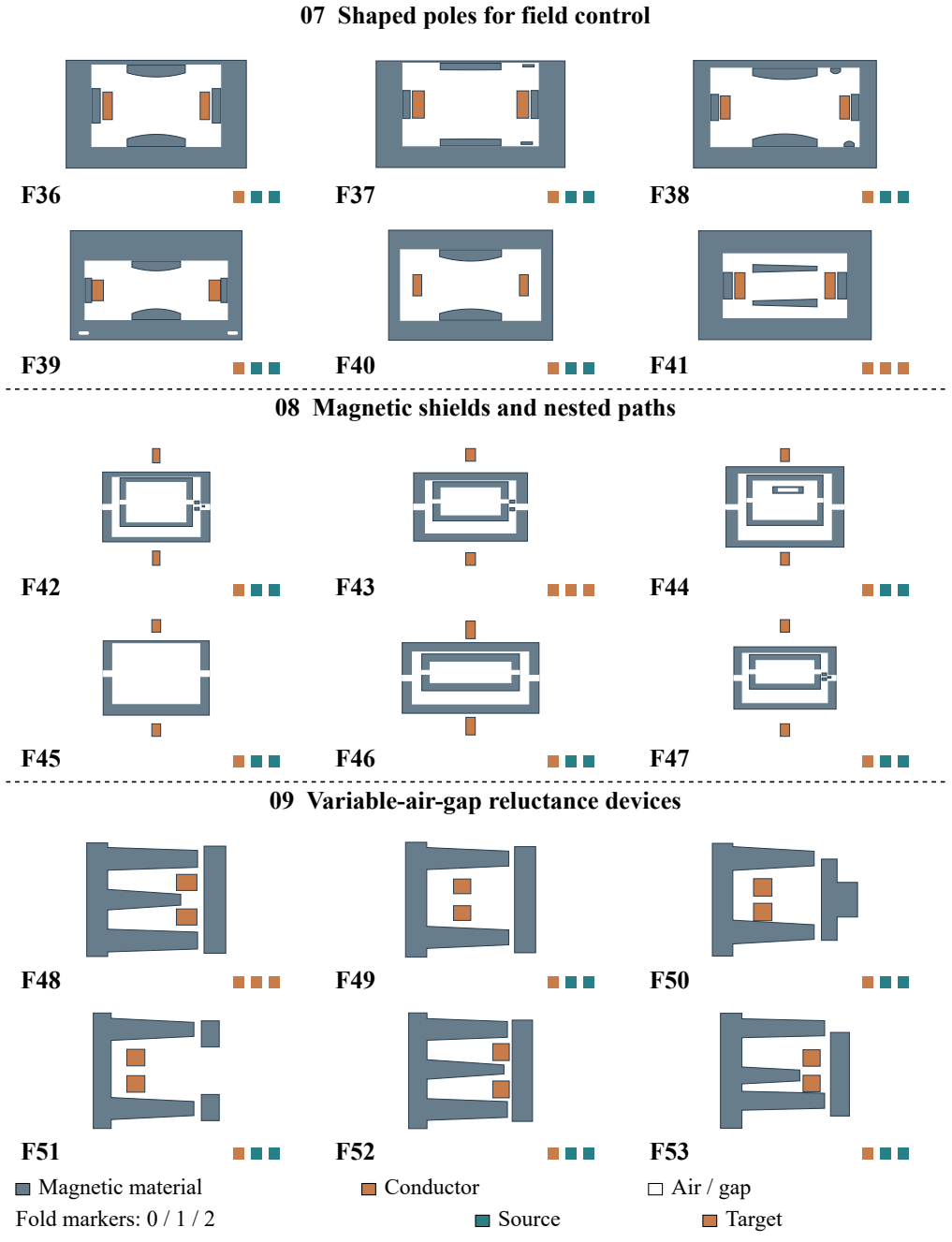


Figure 7: Topology families F36–F53 in EMTopoBench-54, covering shaped poles, shielding and nested structures, and variable-air-gap devices. Material colors, split markers, and scaling conventions are the same as in Figure 5.

C Constitutive Reparameterization Analysis and Initial-Guess Construction

Let $V_h \subset H_0^1(\Omega)$ be a finite element space on the triangular mesh $\mathcal{T}_h$. The weak formulation for the magnetic vector potential is to find $A_{z,h} \in V_h$ such that

$$\int_\Omega \mathcal{H}_m(\mathcal{C}A_{z,h}) \cdot \mathcal{C}v_h \,\mathrm{d}\mathbf{x} = \int_\Omega J_z v_h \,\mathrm{d}\mathbf{x}, \qquad \forall v_h \in V_h. \tag{13}$$

Substituting the basis expansion and applying numerical quadrature gives Equation (11). We next analyze how the material derivatives in this formulation affect prediction error, then describe the field encoding and initial-guess projection.

C.1 Local Error Metric Induced by the Constitutive Mapping

Section 3.2 gives the local relation between H error and weighted B error, which we derive here through the Jacobian of the material mapping. To account for both magnitude and direction, consider a nonzero field within a differentiable interval of the material relation and let $b = \|\mathbf{B}\|$ and $\mathbf{u} = \mathbf{B}/b$, so that $\mathbf{H} = h_m(b)\mathbf{u}$. The magnitude and direction satisfy $\mathrm{d}b = \mathbf{u}^\top \mathrm{d}\mathbf{B}$ and $\mathrm{d}\mathbf{u} = (\mathbf{I} - \mathbf{u}\mathbf{u}^\top)\mathrm{d}\mathbf{B}/b$, respectively. The product rule gives the Jacobian of the material mapping:

$$\mathbf{N} = \mathrm{D}_\mathbf{B}\mathcal{H}_m = \underbrace{h_m'(b)}_{\nu_d}\mathbf{u}\mathbf{u}^\top + \underbrace{\frac{h_m(b)}{b}}_{\nu_s}(\mathbf{I} - \mathbf{u}\mathbf{u}^\top). \tag{14}$$

Evaluating $\mathbf{N}$ at the ground truth, the local error satisfies $\delta\mathbf{H} = \mathbf{N}\delta\mathbf{B} + o(\|\delta\mathbf{B}\|)$, and hence

$$\|\delta\mathbf{H}\|^2 = \delta\mathbf{B}^\top\mathbf{N}^\top\mathbf{N}\delta\mathbf{B} + o(\|\delta\mathbf{B}\|^2). \tag{15}$$

Along the field direction, $\mathbf{N}\mathbf{u} = \nu_d\mathbf{u}$, and perpendicular to the field, $\mathbf{N}\delta\mathbf{B}_\perp = \nu_s\delta\mathbf{B}_\perp$. Orthogonality of the two directions yields Equation (2). This shows that H error distinguishes both material states and error directions, while ν_d and ν_s also enter the material linearization in the FEM tangent matrix.

At zero field, we use the continuous isotropic limit, and at knots in the material table, we use the corresponding one-sided slopes. Each of the three perturbations in Figure 2(b) remains within a single linear segment of the material table, so $\Delta h = \nu_d \Delta b$ agrees with material interpolation. The material Jacobian at these three operating points has also been checked by finite differences.

C.2 Monotonicity and Error Weights of the Invertible Encoding

Training uses encoded fields, so the effect of encoding on the metric above must also be considered. Write the magnitude mapping in Equation (4) as $m_\gamma(r) = s_H r\sqrt{1+r^2}\exp(\gamma r^2/(1+r^2))$, where $h = m_\gamma(r)$ and $r = \|E_\gamma(\mathbf{H})\|$. The Jacobian of the inverse encoding is

$$\mathbf{J}_E = \frac{1}{m_\gamma'(r)}\mathbf{u}\mathbf{u}^\top + \frac{r}{h}(\mathbf{I} - \mathbf{u}\mathbf{u}^\top). \tag{16}$$

We fix the encoding parameter γ for the current input and locally perturb the field prediction. Composing the material mapping with the inverse encoding and including the normalization in Equation (5) gives the local metric of encoding error in B coordinates:

$$\mathbf{G} = \frac{1}{s_z^2}\mathbf{N}^\top\mathbf{J}_E^\top\mathbf{J}_E\mathbf{N}. \tag{17}$$

Here, s_z is the normalization scale of the H encoding loss. The ratio of radial to tangential sensitivity of the inverse encoding is

$$\eta_\gamma(r) = \frac{h}{r m_\gamma'(r)} = \frac{1}{1 + t + 2\gamma t(1-t)}, \qquad t = \frac{r^2}{1+r^2}. \tag{18}$$

The ratio of local weights along and perpendicular to the field direction in the encoding loss is therefore

$$\frac{G_\parallel}{G_\perp} = \left(\frac{\nu_d}{\nu_s}\right)^2 \eta_\gamma(r)^2. \tag{19}$$

The material relation and encoding jointly determine the actual training weights, so the encoded objective cannot be explained solely by mean squared error in physical H. This relation describes how the learning objective responds to different error components, but does not guarantee that these weights reduce Newton iterations.

The encoding must also prevent changes in class from causing a noninvertible or unbounded magnitude correction. Since $|\gamma| \le a = \log(2)/2$, we have $1 + t + 2\gamma t(1-t) \ge 1 - a/2 > 0$, so the magnitude mapping is strictly increasing. For fixed r, the magnitude ratio for two parameter values satisfies $m_{\gamma_1}(r)/m_{\gamma_2}(r) = \exp[(\gamma_1 - \gamma_2)t] \le \exp(2a) = 2$. This bound constrains only the effect of the class-dependent parameters on the same encoded value, not the error in the network prediction itself.

Inverse encoding uses ten scalar Newton root-finding steps, and backpropagation uses the analytical derivative of the radial inverse mapping.

C.3 Material Model, Input Representation, and Supervision

The experiments use a single-valued, isotropic soft magnetic material relation without hysteresis. The material table is sampled from $b(h) = \mu_0 h + J_s \tanh(h/H_0)$, with $J_s = 1.6\,\mathrm{T}$ and $H_0 = 180\,\mathrm{A/m}$. It contains the origin and 63 logarithmically spaced points from 0.018 to $2 \times 10^6\,\mathrm{A/m}$. Label conversion, predicted-field recovery, and FEM all use piecewise linear interpolation of this table, with extrapolation along the slope of the end interval outside the sampled range, keeping the material definition consistent across all three stages.

Geometry and excitation are represented on a 256×256 raster. The seven base channels are air, conductor, and magnetic-material masks, normalized signed current density, two normalized coordinates, and a valid-region mask. To represent the spatial distribution of excitation, seven statistics are computed over windows of 9, 33, and 129 pixels and over the full domain, including positive and negative currents and signed first- and second-order spatial moments. These 28 additional channels and the base channels form a 35-channel input, all computed from known conditions without field labels. Beyond the raster channels, four additional scalars describe the relative distances from the interior modeling domain to the left, right, top, and bottom outer truncation boundaries.

Magnitude labels and scales are determined only from source-family training samples. The H classifier uses the root-mean-square field magnitude in magnetic regions, clusters its logarithm into eight one-dimensional centers, and separates high and low magnitudes at the midpoint of the largest gap between adjacent centers. We set s_H to the exponential of the median log magnitude in the source families and estimate the encoding scale s_z from 20 preselected training samples per source family. The B branch uses source-family scales and classification thresholds for the corresponding split, and these quantities are not refitted during target-family evaluation. The predicted log-relative permeability is bounded by the range allowed by the material table.

The complete direct-supervision objective is

$$\begin{aligned} L_{\mathrm{dir}} = {} & \big\langle \|\mathbf{z}_B - E_{\gamma_B}(\mathbf{B}^*)\|^2 \big\rangle_V + \big\langle \|\mathbf{z}_H - E_{\gamma_H}(\mathbf{H}^*)\|^2 / s_z^2 \big\rangle_M \\ & + \big\langle (\widehat{\ell}_\mu - \ell_\mu^*)^2 \big\rangle_M + 0.1\big[\mathrm{CE}(\mathbf{q}_B, c_B^*) + \mathrm{CE}(\mathbf{q}_H, c_H^*)\big]. \end{aligned} \tag{20}$$

Here, $\langle\cdot\rangle_V$ denotes averaging over valid raster locations, and $\langle\cdot\rangle_M$ denotes averaging within regions with consistent magnetic-material labels, first per sample and then across the batch. The consistency mask affects only the supervised region, while material regions at inference are determined entirely by the input geometry. The remaining notation is the same as in Section 3.3.

Cross supervision additionally compares combinations of the three predictions with the labels:

$$L_{\mathrm{cross}} = \frac{w(e)}{2}\left[\left\langle \frac{\|\widehat{\mu}\widehat{\mathbf{H}} - \mathbf{B}^*\|^2}{s_{B,M}^2} \right\rangle_M + \left\langle \frac{\|\widehat{\mathbf{B}}/\widehat{\mu} - \mathbf{H}^*\|^2}{s_{H,\mathrm{phys}}^2} \right\rangle_M\right]. \tag{21}$$

Both normalization scales are determined from source-family training data. We set $w(e) = 0$ for the first 24 epochs and gradually increase it to 1 over the next 24 epochs. This ablation changes the training constraints but still recovers the magnetic field using the known material table at inference, allowing direct and cross supervision to be compared under the same recovery method.

C.4 Magnetic Field Projection and Extension into Exterior Air

The network predicts fields in the interior modeling domain, while the FEM initial guess must cover the full computational domain, including the exterior air region. Denote these two regions by Ω_{in} and Ω_{out}, respectively. We first fit the candidate field in the interior:

$$\min_{A \in V_h(\Omega_{\text{in}})} \frac{1}{2} \int_{\Omega_{\text{in}}} \|\mathcal{C}A - \widetilde{\mathbf{B}}\|^2 \, \mathrm{d}\mathbf{x}. \tag{22}$$

One interior degree of freedom is temporarily fixed to remove the additive-constant ambiguity in the potential, which is then harmonically extended from the interior boundary into the current-free exterior air region, with zero potential imposed on the outer boundary. The constant in the interior potential does not change the interior magnetic field but does affect the exterior extension, so its value relative to the outer boundary must also be determined.

Let $\mathbf{a}$ be the provisional extension, $\mathbf{e}$ the finite element vector obtained by extending the interior constant 1 into the exterior air region, and $\mathbf{K}_{\text{out}}$ the geometric stiffness matrix of the exterior domain. Using the known load $\mathbf{f}$, the constant correction and final initial guess are

$$c = \frac{\mu_0 \mathbf{f}^\top \mathbf{e} - \mathbf{e}^\top \mathbf{K}_{\text{out}} \mathbf{a}}{\mathbf{e}^\top \mathbf{K}_{\text{out}} \mathbf{e}}, \qquad \mathbf{a}_0 = \mathbf{a} + c\mathbf{e}. \tag{23}$$

This procedure uses only the predicted field, geometry, and known load to convert raster outputs into a discrete initial guess satisfying the potential boundary conditions. Online time includes construction, factorization, and application of the projection matrices.

C.5 Magnitude-Based Routing and Zero Initialization

For samples that are already easy to solve, the cost of constructing a neural initial guess may exceed the iteration savings. Figure 8 illustrates the corresponding zero-initialization path using an unseen sample with an E-shaped core and a planar armature. Its high-H probability is 2.19×10^{-14}, below the common threshold of 0.5, so candidate-field combination and magnetic vector potential projection are skipped, and the solver starts directly from zero potential and meets the acceptance criteria after two Newton steps. This path still includes one network forward pass and changes the initialization cost, not the convergence requirements for the final solution.

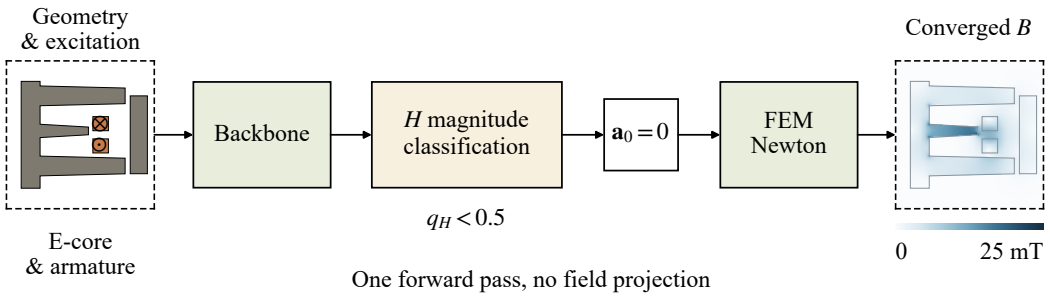


Figure 8: Zero initialization for a low-magnitude sample. The network predicts the magnitude class from the known geometry and excitation. When $q_H < 0.5$, the solver starts directly from zero magnetic vector potential, avoiding candidate-field recovery and projection. The field on the right satisfies the common convergence criteria, with darker blue indicating greater magnitude. Warm gray denotes magnetic material and orange denotes coils.

C.6 Construction of Comparison Methods

All baselines are implemented for the input conditions and FEM evaluation protocol of this study, with the cited literature providing their methodological basis. The architectures, labels, and training budgets follow the common settings used here. All neural prediction methods share a ConvNeXt-Tiny encoder and a UPerNet-based decoder (Liu et al., 2022; Xiao et al., 2018).

Direct-B follows the idea of initializing Newton iteration with neural predictions (Taghikhani et al., 2026), retaining the range-aware encoding and classification supervision for B, and projects the predicted magnetic flux density to an initial guess. It has no auxiliary H or μ heads and does not use H for material-based recovery, so its comparison with CoRaWS examines the effect of the overall output representation scheme. Direct-A_z instead learns the potential field directly, an approach also discussed for magnetic applications in the technical article by Nonnenmacher & Durrande (2026). Its supervision is obtained from Maxwell B fields through a fixed projection and gauge treatment, and after 120 epochs of training, the predicted potential is transferred to FEM degrees of freedom. A separate controlled experiment using FEM potential labels is presented in Appendix D.

POD-ridge-A_z combines a snapshot subspace with data regression and is a nonintrusive reduced-order modeling method (Benner et al., 2015; Kramer & Qian, 2026). After centering source-family potential fields, a rank-32 POD basis is constructed using randomized low-rank decomposition with 8 oversampling vectors (Halko et al., 2011). The input descriptor comprises 16×16 block averages of the seven base channels and four computational domain conditions, giving 1,796 dimensions. After standardization on the source families, ridge regression with an intercept predicts the POD coefficients (Hoerl & Kennard, 1970), using a regularization coefficient of 10^{-4} and no penalty on the intercept. Target-family labels are not used to construct the subspace, standardize inputs, or select hyperparameters.

Following the nearest-neighbor initialization baseline in Sambharya et al. (2024), nearest-neighbor warm starts retrieve a solution by parameter distance from five known support samples in the target family, then initialize FEM through field transfer and projection. This method requires target-family labels and therefore has different data requirements from the zero-shot methods. The cost of solving the support samples is excluded from the subsequent per-sample online time.

D Controlled Experiments on Field Representations and Warm Starts

D.1 Experimental Design

The experiments across topology families involve both structural transfer and output representation. To isolate the effects of the predicted quantity, material-based recovery, and training objective, we conduct controlled comparisons on family F32, an integrated leakage-path topology in the planar segmented flux-diversion magnetic circuit group. This family has 400 training samples and 100 validation samples. Training uses a batch size of 5 for 120 epochs, totaling 9,600 updates. Supervision comes from converged solutions of our FEM solver, with H and μ obtained through the same material relation, so this experiment has different data conditions from the evaluation across families in the main text, which uses Maxwell labels.

All methods are evaluated on the same 20 saturated samples used during method development. The comparisons distinguish two controlled conditions: switching field recovery methods with fixed model weights, and retraining models with different output representations or losses while keeping the training data and budget unchanged. Newton iteration counts and solver acceptance are the primary evidence, while small timing differences provide supplementary evidence.

D.2 Prediction Coordinates, Field Error, and Iteration Counts

Table 5 tests whether switching to H prediction coordinates helps warm starts. For the same directly supervised model, switching from the B head to $H \to B(H)$ reduces median field error from 25.16% to 16.31% and mean Newton iterations from 18.35 to 15.30. To determine whether the gain comes solely from scaling differences, we separately change B to use the same fixed linear scaling and normalized MSE as H in this experiment, but the mean iteration count remains 18.35. Under these controlled conditions, matching the scaling form alone is therefore insufficient to reproduce the gain from H-based recovery.

Cross supervision and recombination of magnitude and direction test whether lower prediction error translates into further iteration savings. Cross supervision reduces field error after H-based recovery to 13.20%, but the mean step count remains close to that of direct supervision. The paired per-case time ratio between the two groups is 0.995, with a resampled 95% interval of $[0.970, 1.020]$, which is

Table 5: Prediction representations and warm-start performance in the single-family experiment on the same 20 development samples. All training groups use epoch-120 parameters, and different recovery methods within a training group reuse the same model weights. Field error is the median relative B error on the raster, and – indicates that an error statistic with the same definition is unavailable.

Training / initial field	B **error (%)** $\downarrow$	**Mean iters.** $\downarrow$
Direct $B/H/\mu$: B head	25.16	18.35
Direct $B/H/\mu$: $H \to B(H)$	16.31	15.30
Cross-supervised $B/H/\mu$: $H \to B(H)$	13.20	15.25
Direct: magnitude from $B(H)$, direction from B	12.69	15.25
Cross-supervised: magnitude from $B(H)$, direction from B	11.05	15.60
Fixed linear B encoding	–	18.35
Native FEM A_z node prediction	–	20.05

insufficient to distinguish their solution speeds. Recovering magnitude from H and taking direction from the B head further reduces field error, but does not consistently reduce iterations for either the directly supervised or cross-supervised model. This shows that better average fits to the same physical field do not necessarily provide better initial states for Newton solves.

We also train a node-query head to predict A_z at quadratic finite element nodes, with joint supervision of the normalized potential and the error in $\mathcal{C}A_z$. Errors are evaluated over the full FEM domain, including exterior air, while Table 5 reports raster-region errors. The median initial B error is 11.58%, compared with 18.09% for H-based recovery, but the mean Newton step count increases to 20.05.

Figure 9 compares the projected initial fields, vector errors, and Newton residual histories of two recovery methods from the same directly supervised model on the first development sample. The B head requires 18 steps, while H-based recovery requires 16.

D.3 Ablations of Material-Based Recovery and Training Objectives

The effects of H prediction, nonlinear material-based recovery, and auxiliary supervision still need to be distinguished in the results above. Table 6 shows that replacing $B(H)$ with a linear mapping while keeping the directly supervised weights fixed increases the mean Newton iteration count to 19.10, indicating a contribution from the known nonlinear material relation in this setting. After removing the auxiliary μ head, the $B + H$ model still requires only 15.60 steps with H-based recovery, fewer than the 18.90 steps needed when directly using the same model's B head. The advantage of H-based recovery therefore persists without auxiliary μ supervision.

Keeping H-based recovery, we next compare the effects of training objectives. Adding recovered-field error or a magnetic-energy Bregman loss to direct H prediction yields 15.65 and 15.55 steps, respectively, neither below the 15.30 steps from direct supervision. The two loss coefficients are set in advance using gradient norms on training samples, without tuning to solver performance on the development samples. Although these objectives improve raster prediction error, iteration counts do not improve with them, again showing a gap between field-fitting and solver-acceleration objectives.

Using Direct-B as another baseline, we add field error after one Newton update to the training objective to test learning an initial guess directly through the iteration process (Sambharya et al., 2024). The base B supervision covers all 400 training samples, and the additional term is computed twice per epoch on eight preselected saturated training samples, for a total of 240 evaluations. With this limited physics-supervision budget, the model requires 18.50 steps on average.

The nearest-neighbor method uses existing solutions directly, without training a network. Initial guesses retrieved by parameter distance from all 400 training solutions require 22.75 steps. The nearest-neighbor baseline in the main text uses five support samples, compared with the 400 training solutions available in this within-family experiment. All evaluations are performed sequentially after training has finished.

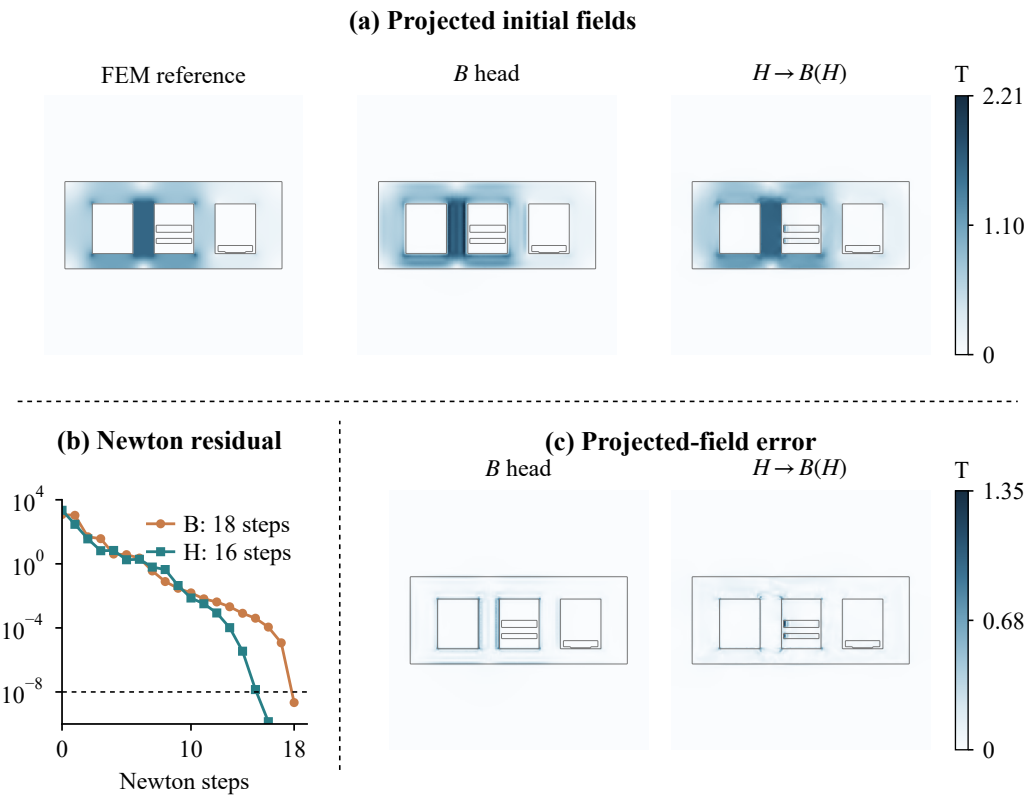


Figure 9: Initial guesses and Newton iterations for two field recovery methods with fixed weights. (a) Field magnitudes of the FEM reference on the same mesh and the two projected initial guesses. (b) Relative discrete residuals from each initial guess, with the dashed line indicating the 10^{-8} convergence threshold. The B head and H-based recovery require 18 and 16 steps, respectively. (c) Magnitudes of the vector errors in the initial fields relative to the reference, with gray lines marking magnetic-material boundaries. The field plots show the spatial error distribution, and the residual curves show the corresponding solution process.

Table 6: Ablations of material-based recovery, auxiliary supervision, and training objectives. Models are trained for 120 epochs on the same topology family and solve the same 20 development samples. Accepted / trials reports accepted solves and total attempts, with 20 and 40 attempts corresponding to one and two evaluations per case, respectively.

Training / deployment	**Epoch**	**Mean iters.**	**Accepted / trials**
Direct $B/H/\mu$: H with a fixed linear map	120	19.10	20/20
$B + H$ supervision: $H \to B(H)$	120	15.60	20/20
$B + H$ supervision: B head	120	18.90	20/20
$B + \log \mu$ supervision: B head	120	18.75	20/20
Direct H route + recovered-field error	120	15.65	40/40
Direct H route + constitutive energy loss	120	15.55	40/40
Direct-B + field error after one Newton step	120	18.50	40/40
Nearest neighbor among 400 training solutions	–	22.75	40/40

D.4 Supplementary Comparison with Validation-Based Model Selection

The preceding comparisons use fixed epoch-120 weights. To test whether the advantage of the recovery method depends on this choice, Table 7 instead selects models by validation error, then switches the recovery method while keeping the selected model's weights fixed. For the same cross-supervised model at epoch 110, directly using B, using the predicted μH, and recovering through the material table require 18.30, 17.00, and 15.30 steps, respectively. The advantage of material-based recovery over directly reading the output heads therefore also appears in this model.

Table 7: Comparison of recovery methods with models selected by validation error. Each group is trained for 120 epochs, and Epoch denotes the epoch of the selected model. Different recovery methods within the same training group and epoch share weights, and each row evaluates the 20 development samples twice. The final row repeats evaluation after reconstructing initial guesses from the same model.

Training / deployment	**Epoch**	**Mean iters.**	**Accepted / trials**
Direct-B: B head	110	18.95	40/40
Direct $B/H/\mu$: B head	120	18.35	40/40
Direct $B/H/\mu$: μH	120	22.95	40/40
Cross-supervised: B head	110	18.30	40/40
Cross-supervised: μH	110	17.00	40/40
Cross-supervised: $H \to B(H)$	110	15.30	40/40
Cross-supervised: $B/\mu \to B(H)$	110	16.30	40/40
Cross-supervised: μH, repeated evaluation	110	17.05	40/40

D.5 Sensitivity Analysis Excluding F32

F32 was used in the single-family method-development experiments. To assess its influence on the main results, we exclude all 20 evaluation pairs from this family and aggregate the existing results for the remaining 1,420 pairs. The trained models, routing decisions, and timing convention remain unchanged. CoRaWS reduces cumulative online time from 18.6087 h to 17.0720 h, a reduction of 8.26%, and total Newton updates from 7,260 to 6,421, a reduction of 11.56%. Both CoRaWS and the zero-initialization baseline satisfy final acceptance on all remaining pairs. Thus, the aggregate benefit persists after excluding the development family from evaluation.