CONSERVATIVE COMPOSITIONAL ROUTING: STRUCTURING LATENT-STATE FORMATION FOR ZERO-SHOT COMPOSITIONAL GENERALIZATION IN NEURAL PDE SOLVERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural PDE solvers provide approximate solutions to partial differential equations significantly faster than traditional numerical methods and can provide accurate predictions for new geometries or changes in physical parameters. However, a question remains less explored: what if a model has seen different physical constituents with different material properties separately during neural training, but has never seen these constituents combined within the same heterogeneous domain? We aim to improve zero-shot compositional generalization in heterogeneous PDEs by incorporating material and interface information into the formation of latent physical states. We introduce a Material-Interface Guided Transolver with Conservative Compositional Routing (CCR-MIG), in which material information guides how mesh points are assigned to the learned latent states through a bounded weighting factor that limits how strongly the material information can modify the original routing. A conflict-aware gate further reduces the material contribution in regions where the influences of different constituents overlap. We evaluate the proposed routing strategy on two heterogeneous PDE problems: elasticity and steady thermal conduction. The results show that CCR-MIG reduces the mean relative error of the original Transolver from 0.07134 to 0.05891 for elasticity and from 0.03494 to 0.02324 for thermal conduction. We also compare CCR-MIG with a feature-matched Transolver that receives the same material information but does not use it to modify latent-state routing. We observe that CCR-MIG improves over this feature-matched baseline in both cases, reducing the error by 23.4% for elasticity and 18.9% for thermal conduction. These results show that simply providing constituent information as an additional input is not sufficient to achieve consistent compositional generalization, and that how this information is introduced into latent-state formation is also important.

1 INTRODUCTION

Heterogeneous and architected materials derive their effective behavior from the arrangement, properties, and interactions of their constituent phases, which lets stiffness, toughness, and thermal transport be tuned through microstructural design (Gao et al., 2003; Sharma et al., 2023; Bertoldi et al., 2017; Schittny et al., 2013; Damavandi et al., 2022a; 2025; 2022b). Because analytical solutions are rarely available for such heterogeneous domains, predicting their response typically relies on finite element analysis (FEA) (Reddy, 1993). However, repeated high-fidelity simulations become expensive in design exploration, inverse analysis, and uncertainty quantification (Bessa et al., 2017), especially for heterogeneous constituent materials. Surrogate modeling offers remarkable potential in developing computationally efficient frameworks (Raissi et al., 2019; Karniadakis et al., 2021; Azizzadenesheli et al., 2024); among them, transformer-based PDE solvers (Cao, 2021; Li et al., 2023a; Hao et al., 2023; Li et al., 2023b; Zhao et al., 2024), building on the Transformer architecture (Vaswani et al., 2017), have shown strong potential for PDE modeling. In particular, Transolver introduces Physics-Attention (Wu et al., 2024), which softly assigns mesh nodes to a compact

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.

set of latent physical-state slices. Transolver-based models have since been scaled to million- and industrial-scale geometries (Luo et al., 2025; Zhou et al., 2026) and extended with geometric context (Adams et al., 2025). While existing transformer solvers have made significant strides in scalability and geometric representation, they do not address a different form of distribution shift: a change in the *co-occurrence* of familiar material constituents. This poses a major challenge: can a neural PDE solver generalize when familiar physical constituents appear together in combinations never observed during training? For instance, a training set may contain each constituent class separately: an elasticity model may see domains with either soft or hard inclusions and later encounter both; similarly, a thermal model may see low- and high-conductivity inclusions separately, and during inference the model may encounter mixed constituents. We call this setting *zero-shot compositional generalization*; "zero-shot" refers to the absence of the mixed composition from training, not of the individual constituents. The model does not lack information about the constituents; it lacks a learned rule for composing that information when they coexist. For Physics-Attention, material information can enter the model at two levels. The first is content conditioning. Here, the model captures material information by incorporating material descriptor fields in the nodal features. This gives the model access to constituent information, but its effect on how nodes are grouped into latent slices is only indirect. The second, routing conditioning, lets the model construct a representation by incorporating material information at the slice-assignment level. This acts directly on how the latent physical states are formed; we bound this intervention so that it cannot override the learned routing. We study the impact of routing-level material conditioning on the model's zero-shot compositional capability.

Motivated by these challenges, we introduce a *Material–Interface Guided Transolver with Conservative Compositional Routing* (CCR-MIG). In CCR-MIG, constituent and interface information serve as input features to guide the formation of latent physical states, and a bounded weighting factor is combined with the original Transolver routing, limiting how much the original Transolver routing can change. In addition, a conflict-aware gate further reduces the material contribution in regions where the influences of different constituents overlap. We compare three models: the original Transolver, a feature-matched Transolver that receives the same constituent-influence features but retains the original routing, and CCR-MIG. All models use the same architecture and hyperparameters and are evaluated on heterogeneous elasticity and steady thermal conduction problems with the same microstructures, material regions, and meshes. The results indicate that CCR-MIG improves over both the original Transolver and the feature-matched baseline for both PDE problems. Our contributions are:

1. We propose a zero-shot compositional generalization setting for heterogeneous partial differential equations (PDEs) in which each constituent class is seen separately during training and the classes appear together only at inference.
2. We distinguish between content and routing conditioning and provide a baseline to study the effect of modifying latent state representations.
3. We introduce conservative compositional routing, which defines a convex combination of constituent-specific routing and data-driven routing and provides an explicit ℓ_1 bound on the deviation from the data-driven routing.
4. CCR-MIG improves zero-shot compositional generalization over both Transolver and the feature-matched baseline on both PDEs.

2 Related Work

Operator learning on structured representations DeepONet learns solution operators through a branch–trunk decomposition (Lu et al., 2021; He et al., 2023; 2024), while the Fourier Neural Operator (FNO) learns operators using Fourier-space representations (Li et al., 2021; Tran et al., 2023). Both are widely used operator-learning approaches (Kovachki et al., 2023). FNO and many related neural operators are commonly developed on regular discretizations, which makes their direct application to variable-size unstructured meshes with sharp material interfaces less straightforward. Graph-based simulators (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021; Brandstetter et al., 2022) and GINO (Li et al., 2023c) are better suited to irregular meshes and complex geometries. However, existing benchmarks such as PDEBench (Takamoto et al., 2022) mainly test generalization to new physical parameters or discretizations. In our problem, the main challenge is inference when mul-

tiple constituents appear together, while the constituent classes are only observed separately during training.

Attention-based solvers and latent physical states Another paradigm uses Transformer solvers, such as Galerkin-type attention (Cao, 2021), OFormer (Li et al., 2023a), and GNOT, whose geometric gating acts as a soft domain decomposition (Hao et al., 2023), to approximate solution operators with attention. Li et al. (2023b) reduce attention cost for PDE surrogates and Zhao et al. (2024) combine attention with physics-informed training. Transolver (Wu et al., 2024) replaces attention between individual mesh nodes with attention over learned physical-state slices. Later studies extend this framework to million-scale (Luo et al., 2025) and industrial-scale geometries (Zhou et al., 2026), as well as geometric encoding (Adams et al., 2025). Among these methods, GeoTransolver is the most related to our work. It introduces multi-scale geometric information into Transolver by adding local geometric features to the point representation and passing global geometric information to the slice representations. However, the node-to-slice assignment still follows the original Transolver mechanism, so the geometric information affects the routing indirectly. We take a different approach by using material information directly at the node-to-slice routing level and limiting how much this material-guided routing can change the original data-driven routing. GeoTransolver mainly considers generalization across geometry and boundary conditions, whereas our work considers generalization when familiar material constituents appear together in combinations that were not present during training.

Heterogeneous and multiscale material modeling Classical homogenization relates constituent properties to effective response through representative volume elements (Hill, 1963; Geers et al., 2010; 2017; Pavliotis & Stuart, 2008; Charalambakis, 2010; Terada et al., 2000), with extensions to non-periodic media (Fish & Wagiman, 1993), general multiscale coupling (Attinger & Koumoutsakos, 2004), and data-driven variants for discrete biological materials (Kumar et al., 2025). These methods encode the intuition we exploit, that constituent identity organizes the response, but assume the constituent statistics at evaluation resemble those used to build the model. Our reference solutions are produced directly by finite element analysis (Reddy, 1993; Cook et al., 2001; Hughes, 2000), meshed with Gmsh (Geuzaine & Remacle, 2009) and solved with JAX-FEM (Xue et al., 2023).

Compositional generalization Compositional generalization studies whether a model can use components seen during training in new combinations at test time (Lake & Baroni, 2018; Keysers et al., 2020; Hupkes et al., 2020), and relational inductive biases have been explored for this purpose (Battaglia et al., 2018). In neural PDE solvers, out-of-distribution studies have focused on changes in geometry, discretization, or physical parameters (Zhou et al., 2025; Lara Benitez et al., 2024). In our approach, the model is trained on material constituents appearing separately and is evaluated when multiple constituents appear together. The major challenge is combining previously learned information about the individual constituents.

3 Problem Setting and Benchmarks

3.1 Zero-Shot Compositional Generalization

This study considers two heterogeneous PDE problems in a matrix with embedded inclusions. For elasticity, the inclusions are classified as soft or hard based on their stiffness relative to the matrix. For thermal conduction, the corresponding constituents are low- and high-conductivity inclusions relative to the matrix.

Benchmarks For the elasticity problem, we use a rectangular matrix domain $\Omega = [0, 10] \times [0, 5]$ with $E_m = 10^5$ and $\nu = 0.3$. We place circular inclusions inside the matrix and sample the stiffness ratio from $E_s/E_m \in [0.05, 0.5]$ for soft inclusions and $E_h/E_m \in [2, 20]$ for hard inclusions. The elasticity problem is solved under displacement-controlled uniaxial loading. For model training, samples contain either soft or hard inclusions, but never both. They contain either one inclusion or two to three inclusions of the same class. For inference, a mixed evaluation set is created in which one soft and one hard inclusion are placed in the same domain using the same material property ranges and a closely matched inclusion-size range. For thermal conduction, we use the same microstructures and meshes as in the elasticity problem. Temperature is prescribed on the left and right boundaries, while the top and bottom boundaries are insulated. The elastic modulus contrast is

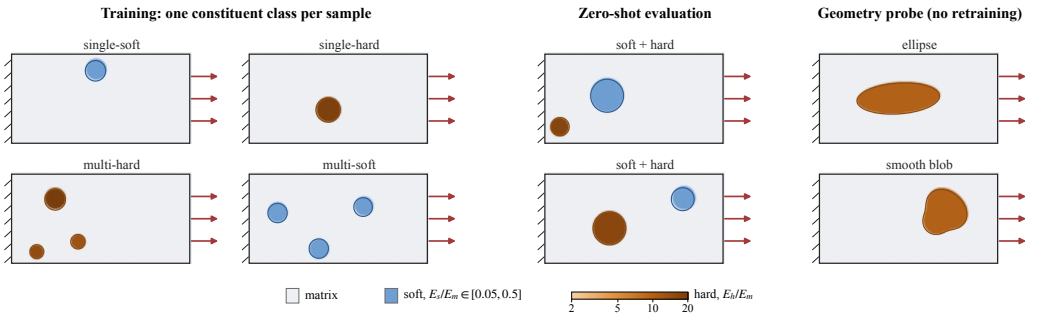

Figure 1: Compositional split. Left: training samples with a single constituent class. Middle: zero-shot evaluation with soft (blue) and hard (orange) inclusions in the same plate. Right: non-circular inclusions for the geometry probe. Plates are clamped on the left and loaded on the right.

mapped to thermal conductivity using

$$k(\mathbf{x})/k_m = E(\mathbf{x})/E_m.$$

For training, a sample contains either low- or high-conductivity inclusions, while during evaluation both classes appear together in the same domain. We call this inference problem zero-shot compositional generalization. Using the same geometry and mesh for both problems allows us to study the same compositional setting under two different governing PDEs. Figure 1 shows representative training and inference samples for the elasticity and thermal PDEs. Appendices A and B provide the corresponding finite element formulations.

4 Method

This section discusses the general methodology of the proposed CCR-MIG architecture. This study employs constituent information in Physics-Attention through two complementary pathways: First, material influence fields are added to the nodal input features. Second, it uses the same information in Conservative Compositional Routing to modify how nodes are assigned to latent slices. The routing modification is bounded so that the original data-driven routing is still retained. Figure 2 shows the method's overall structure. We first describe the Physics-Attention backbone, followed by these two components.

4.1 Physics-Attention Backbone

Physics-Attention (Wu et al., 2024) groups N mesh nodes into $K \ll N$ latent physical-state slices and applies self-attention before mapping the updated information back to the mesh. It reduces the attention cost from $\mathcal{O}(N^2)$ to $\mathcal{O}(NK + K^2)$, for details see (Appendix C). In our proposed model, we modify how the nodes are assigned to these latent slices while leaving the subsequent self-attention and mapping steps unchanged. For node i, Transolver first computes the routing logits $\mathbf{s}_i \in \mathbb{R}^K$ and obtains the original routing distribution as

$$\mathbf{p}_i^{\text{free}} = \text{softmax}\left(\frac{\mathbf{s}_i}{\tau}\right), \tag{1}$$

where τ is the routing temperature. The remaining steps of Physics-Attention are unchanged.

4.2 Content Conditioning: Continuous Constituent-Influence Fields

For each node i, we construct continuous influence fields for the two constituent classes, $a_i^s \in [0, 1]$ and $a_i^h \in [0, 1]$, strongest inside the corresponding material regions and decaying away from them. Concretely, for $c \in \{s, h\}$,

$$a_i^c = \min\left\{1, \left(\frac{r_c}{r_c + \eta d_{i,c}}\right)^2\right\}, \qquad \eta = 0.6, \tag{2}$$

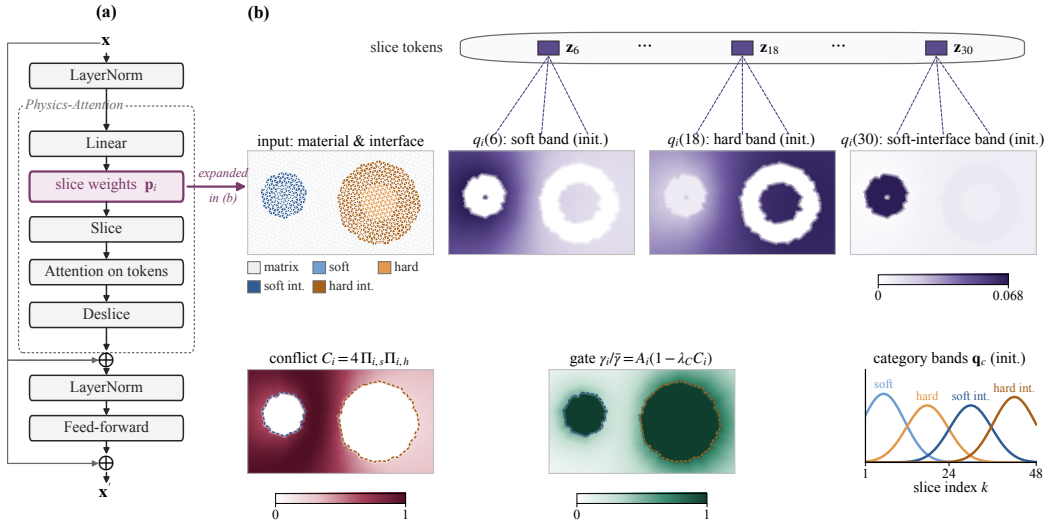

Figure 2: Overview of CCR-MIG. (a) CCR-MIG layer, with the modified slice-weight module highlighted. (b) Routing on a held-out elasticity sample: material input, initialized routing priors $q_i(k)$ and category bands $\mathbf{q}_c$, and trained-model conflict C_i and gate $\gamma_i/\bar{\gamma}$ (Section 4.3).

where $d_{i,c}$ is the distance from node i to the nearest node of constituent c (inclusion or interface) and r_c is an equivalent radius estimated from the area of that constituent (Appendix E.2). Continuity matters: a binary phase indicator is discontinuous across interfaces, whereas a decaying field also expresses proximity to one. The conditioned nodal representation is

$$\widetilde{\mathbf{f}}_i = \left[\mathbf{f}_i,\ a_i^s,\ a_i^h,\ a_i^s + a_i^h\right]. \tag{3}$$

We do not include the interaction term $a_i^s a_i^h$ because soft and hard constituents never appear together during training. Therefore, the model has no training samples to learn this interaction, and the term would only activate for mixed-composition samples. Both the feature-matched baseline and CCR-MIG use the same nodal features in Eq. (3); they differ only in the routing mechanism.

4.3 Conservative Compositional Routing

A simple way to include material information in the routing is to add material-dependent terms directly to the routing logits. However, this can change the original routing by an uncontrolled amount. Instead, CCR builds a separate material-guided routing distribution and combines it with the original Transolver routing.

We use four routing categories, $c \in \mathcal{R} = \{s,\ h,\ s\text{-int},\ h\text{-int}\}$, corresponding to soft inclusion, hard inclusion, soft interface, and hard interface. Each category has a learnable distribution over the K latent slices:

$$q_c(k) = \frac{\exp\left[-\dfrac{(k-\mu_c)^2}{2\sigma_c^2\tau_m}\right]}{\sum_{j=1}^{K}\exp\left[-\dfrac{(j-\mu_c)^2}{2\sigma_c^2\tau_m}\right]}, \tag{4}$$

where μ_c and σ_c define the center and width of the slice distribution, and τ_m is the material-routing temperature.

Let $a_{i,c}$ denote the routing activity of category c at node i. Nodes within a band of half-width $2h$ around an inclusion boundary ($h = 0.15$ is the nominal mesh size) are assigned exclusively to the soft- or hard-interface category; all other nodes carry a_i^s and a_i^h in the soft- and hard-inclusion categories (Appendix E.3). The activities are normalized as

$$\pi_{i,c} = \frac{a_{i,c}}{\sum_{c'} a_{i,c'} + \epsilon}, \qquad \mathbf{q}_i = \sum_c \pi_{i,c}\mathbf{q}_c. \tag{5}$$

The final routing distribution is obtained by mixing the original Transolver routing with the material-guided routing:

$$\mathbf{p}_i = (1-\gamma_i)\mathbf{p}_i^{\text{free}} + \gamma_i \mathbf{q}_i, \qquad 0 \le \gamma_i \le \gamma_{\max}. \tag{6}$$

The factor γ_i controls how much the material-guided routing changes the original routing. We also reduce this contribution when the soft and hard influences overlap. We define

$$\Pi_{i,s} = \pi_{i,s} + \pi_{i,s\text{-int}}, \qquad \Pi_{i,h} = \pi_{i,h} + \pi_{i,h\text{-int}}, \tag{7}$$

and use

$$C_i = 4\,\Pi_{i,s}\Pi_{i,h}, \tag{8}$$

where C_i is small when one constituent is dominant and becomes larger when the two influences are similar.

The routing gate is then

$$\gamma_i = \bar{\gamma} A_i (1 - \lambda_C C_i), \qquad \bar{\gamma} = \gamma_{\max}\sigma(g), \tag{9}$$

where

$$A_i = \min\left(1, \sum_c a_{i,c}\right).$$

Here, g is a learnable scalar (one per layer and attention head), σ is the sigmoid, λ_C sets the strength of conflict suppression, and γ_i is clipped to $[0, \gamma_{\max}]$. A_i reduces the routing modification away from the material regions, while C_i reduces it when the soft and hard influences overlap.

Bounded intervention CCR modifies the original Transolver routing through the mixing factor γ_i. Since both $\mathbf{p}_i^{\text{free}}$ and $\mathbf{q}_i$ are probability distributions, their L_1 distance cannot exceed 2. Therefore,

$$\left\|\mathbf{p}_i - \mathbf{p}_i^{\text{free}}\right\|_1 = \gamma_i \left\|\mathbf{q}_i - \mathbf{p}_i^{\text{free}}\right\|_1 \le 2\gamma_i \le 2\gamma_{\max}. \tag{10}$$

Thus, $\gamma_{\max}$ directly limits how much the material-guided routing can change the original routing. When no constituent influence is present, $A_i = 0$ and the routing remains unchanged.

4.4 Training Objective

To compare the model prediction $\widehat{\mathbf{Y}}$ with the reference solution $\mathbf{Y}$, we use the global relative field error:

$$\mathcal{L}_{\text{rel}} = \frac{\|\widehat{\mathbf{Y}} - \mathbf{Y}\|_2}{\|\mathbf{Y}\|_2 + \epsilon}, \tag{11}$$

where $\mathbf{Y} = \mathbf{U} = [u_x, u_y]$ for elasticity and $\mathbf{Y} = T$ for thermal conduction. No auxiliary loss terms are used. Therefore, the feature-matched baseline and CCR-MIG are trained using the same supervision.

5 Experimental Setup

5.1 Training, Development, and Final Test Protocol

Each non-mixed dataset contains 400 samples, with 360 used for training and 40 for validation. Each sample contains 2754–2900 nodes and 5304–5596 triangular elements. The dataset construction and the different sample types are summarized in Figure 1. The training and evaluation protocol is shown in Figure 3. Models are trained on non-mixed samples, while mixed samples are reserved for development and final testing. Only non-mixed training samples are used to update model parameters. For elasticity, we compared routing formulations on 100 mixed development samples (DEV) and selected conservative compositional routing from these comparisons. The results are described in the ablation study in Section 6.1. We then fixed the routing formulation and hyperparameters and evaluated the models on 200 independently generated mixed test samples (FINAL). This test set had not been inspected during development or used for model selection. For thermal conduction, we trained the models on non-mixed thermal samples using the same network architecture, training settings, and routing hyperparameters as in elasticity. A separate set of 100 mixed thermal development samples was used for the routing ablation reported in Section 6.1 and Appendix G, but was not used to select routing variants, hyperparameters, or training settings. Final testing used a separate set of 200 independently generated mixed samples.

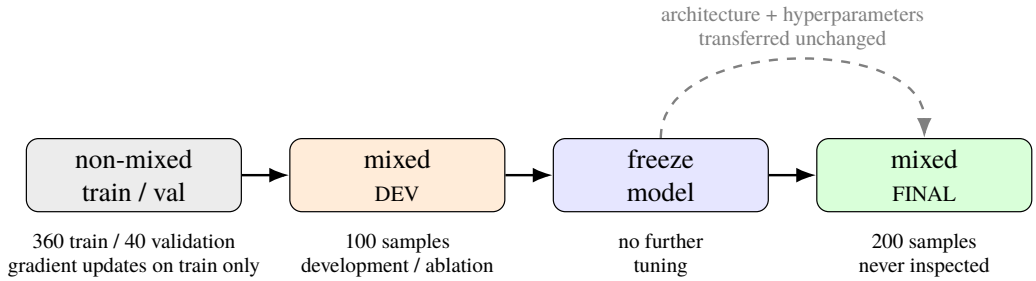

Figure 3: Training, development, and final-test protocol for both PDEs.

5.2 Compared Models

We compare three models with the same backbone dimensions and training settings. Transolver (Wu et al., 2024) uses the original nodal descriptors and data-driven routing. Feature-matched Transolver uses the original Transolver routing with the augmented input $\widetilde{\mathbf{f}}_i = [\mathbf{f}_i, a_i^s, a_i^h, a_i^s + a_i^h]$ from Eq. (3). The two influence fields, a_i^s and a_i^h, represent soft and hard inclusions for elasticity and low- and high-conductivity inclusions for thermal conduction. Section 3 gives the corresponding material parameters and ranges. CCR-MIG uses these same features with Conservative Compositional Routing. The three models allow us to assess the two changes in turn: adding material features to Transolver, then using those features in the routing to adjust node assignments to latent slices.

5.3 Training Configuration

All models use six Physics-Attention layers, 48 latent slices, a hidden dimension of 192, and eight attention heads, with no dropout and approximately 1.22M trainable parameters. Models are trained for 1000 epochs using AdamW (Loshchilov & Hutter, 2019), with an initial learning rate of 10^{-3} and cosine learning-rate decay (Loshchilov & Hutter, 2017) without warmup. Weight decay is set to 10^{-5}, and the gradient norm is clipped at 1.0. The batch size is one because the meshes differ in node count. The routing hyperparameters are set to $\gamma_{\max} = 0.30$, a gate initialization value of 0.05, a conflict suppression coefficient of $\lambda_C = 0.75$, and a material-routing temperature of $\tau_m = 1.0$. By Eq. (10), the ℓ_1 distance between the adjusted routing distribution and the original Transolver routing distribution does not exceed 0.6. Each model is run with six random seeds, $\{123, 456, 789, 1011, 2022, 3033\}$. All experiments are conducted on a single NVIDIA RTX PRO 5000 GPU. The full configuration is provided in Appendix E.

5.4 Evaluation Metrics and Statistical Treatment

We evaluate displacement and temperature predictions on the final test set using the relative field error in Eq. (11), denoted by E_u and E_T, respectively. For each seed, we average the errors over the 200 test samples. We report the mean and sample standard deviation of the resulting six values. All models share the same data split and test set. For each model pair, we calculate the difference in average error for each seed and use these six differences in an exact two-sided sign-flip test. We also report Cohen's d_z (Cohen, 1988) for the paired differences and apply Holm correction (Holm, 1979) to the four comparisons in Appendix F.2. Predicted fields are plotted for visual comparison; the relative field error remains the primary metric.

6 Results

For the elasticity and thermal PDEs, we use the same architecture and routing hyperparameters to maintain consistency. We evaluated the performance of the proposed CCR-MIG on the held-out test sets using the relative L_2 field error. Table 1 compares the three models on the mixed-composition test sets. For elasticity, the errors were 0.07134 for Transolver, 0.07686 for the feature-matched model, and 0.05891 for CCR-MIG. For thermal conduction, the corresponding errors were

Table 1: Zero-shot compositional generalization on the held-out 200-sample mixed-composition benchmarks. Mean ± sample standard deviation of the per-seed mean relative error over six seeds. Reduction is relative to Transolver.

	Elasticity		Thermal	
Model	E_u	Red.	E_T	Red.
Transolver	0.07134 ± 0.02029	–	0.03494 ± 0.02045	–
Feature-matched Transolver	0.07686 ± 0.02831	-7.7%	0.02865 ± 0.00730	18.0%
CCR-MIG	$\mathbf{0.05891 \pm 0.01187}$	$\mathbf{17.4\%}$	$\mathbf{0.02324 \pm 0.00356}$	$\mathbf{33.5\%}$

Table 2: Routing ablations on the 100-sample mixed-composition development benchmark (six-seed mean relative error). Direct material routing adds constituent biases to the logits without the convex mixture; removing the conflict gate keeps the mixture but applies $\gamma_i = \bar{\gamma} A_i$ uniformly.

Variant	Elasticity	Thermal
Feature-matched (no routing intervention)	0.07355	0.02634
Direct material routing (unbounded)	0.08700	0.02676
CCR without conflict gate	0.05863	**0.02120**
CCR-MIG (full)	**0.05454**	0.02141

0.03494, 0.02865, and 0.02324. Compared with Transolver and the feature-matched model, CCR-MIG reduced the error by 17.4% and 23.4% for elasticity, and by 33.5% and 18.9% for thermal conduction. Paired effect sizes favor CCR-MIG in all four comparisons ($d_z = 0.57$–0.64), but with six seeds none is significant after Holm correction ($p_{\text{adj}} = 0.75$; Appendix F.2); we therefore read these results as a consistent trend with reduced across-seed variance rather than a statistically conclusive separation.

6.1 Routing Ablations

Table 2 compares different variants of the proposed routing method on the mixed-composition development set. Direct material routing gives higher error than the feature-matched model for both PDEs. In elasticity, the error changes from 0.07355 to 0.08700. After using the bounded convex mixture, the error decreases to 0.05863 for elasticity and 0.02120 for thermal conduction. With the introduction of conflict gate routing, the elasticity error further decreases to 0.05454. For thermal conduction, the error remains almost the same, changing from 0.02120 to 0.02141. The difference also changes sign across seeds ($d_z = -0.04$, $p = 0.94$; for more detail, see the ablation study in Appendix G). We further visualize the routing behavior in Figure 4. Inside and around the inclusions, C_i is generally small because one routing category has a larger contribution. When the soft and hard influence fields overlap, C_i increases and the contribution from material-guided routing is reduced. Figure 4 shows γ_i, C_i, and the change in routing, $\|\mathbf{p}_i - \mathbf{p}_i^{\text{free}}\|_1$, for a representative mixed-composition sample.

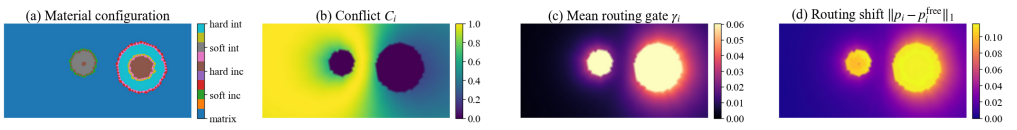

Figure 4: Routing diagnostics on a FINAL-200 elasticity sample: (a) routing categories, (b) conflict C_i, (c) mean gate γ_i averaged over layers and attention heads, and (d) routing shift $\|\mathbf{p}_i - \mathbf{p}_i^{\text{free}}\|_1$.

6.2 Qualitative Predictions on Zero-Shot Mixed Composition

In this section, we present qualitative CCR-MIG field predictions for zero-shot mixed-composition samples for both PDEs. For elasticity, we compare the predicted displacement field with the FEM

reference displacement field. For thermal conduction, we compare the predicted temperature field with the FEM reference temperature field. The prediction error is evaluated using the relative L_2 field error, and the mean error over the test samples is used for the quantitative comparison. Figure 5 shows four representative zero-shot examples, including two elasticity samples and two steady thermal-conduction samples. CCR-MIG reproduces the overall field patterns and the effects of material contrast well. Most of the field error occurs near material interfaces and around the inclusions, where the solution changes more rapidly. These examples show the model predictions for mixed-composition cases that were not present during training.

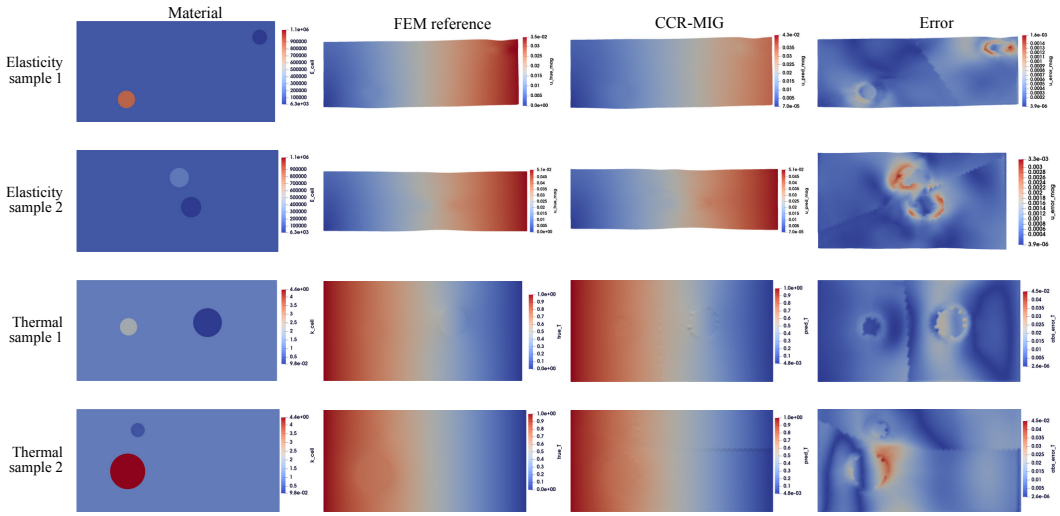

Figure 5: Zero-shot mixed-composition predictions for elasticity (top) and thermal conduction (bottom).

7 Conclusion

To address zero-shot compositional generalization in heterogeneous PDEs, this work develops CCR-MIG, a neural PDE solver that brings material information into the latent-state formation process of Transolver. Instead of using constituent information only as extra input features, CCR-MIG also uses it to guide how mesh nodes are assigned to latent slices. At the same time, the change from the original Transolver routing is kept limited. A conflict-aware gate is also introduced for regions where the influence of different constituents overlaps. The results show that simply giving the network more material information does not always help. In some cases, adding the constituent information only to the input features improves the prediction, while in other cases it makes the result worse. The main improvement comes when the same information is used in the routing process. Even then, directly forcing the routing toward the material information is not enough. The ablation study shows better results when the material-guided change is limited and the original data-driven routing is still preserved. Based on this result, we describe the main idea behind CCR-MIG as bounded intervention. The material information is allowed to influence the routing, but not to completely determine it. In other words, the prior information is used to guide what the model has already learned rather than replace it. This allows the solver to use information learned from individual constituents when those constituents appear together during evaluation, even though that combination was not present during training. There are still several directions to explore. The same approach can be tested on more complex geometries, three-dimensional problems, and other PDE systems. It would also be useful to study whether bounded intervention can be applied to other neural operator architectures and to problems with more material constituents and more complicated interactions. A further direction is a gate that adjusts its own conservatism according to how far a test composition lies from those seen during training. These extensions could help determine how general this idea is for predicting new heterogeneous systems formed from constituents that the model has already seen separately.

AI USE STATEMENT

Generative AI tools were used for language polishing, editing, and improving the clarity and organization of portions of the manuscript. All technical content, experimental design, datasets, analyses, results, interpretations, and conclusions were produced and verified by the authors. Any AI-suggested references were independently checked against the original sources before inclusion. The authors take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work studies neural PDE surrogate modeling using synthetically generated heterogeneous microstructures and finite element simulation data. It does not involve human subjects, personally identifiable information, or sensitive datasets. We are not aware of specific ethical risks associated with the datasets or experiments presented in this study. The authors have reviewed the ICLR Code of Ethics and take responsibility for the accuracy and integrity of the reported methodology and results.

REPRODUCIBILITY STATEMENT

Section 4 describes the CCR-MIG architecture, constituent-influence representation, routing mechanism, and training objective. Section 5 specifies the training, development, and final evaluation protocols, including the separation between the development and final mixed-composition datasets and the routing hyperparameters used. Appendices A and B give the finite element formulations used to generate reference solutions, and Appendix C summarizes the Physics-Attention backbone. Additional implementation details, seed-wise results, and ablation experiments are provided in Appendices E–G. Code for dataset generation, training, and evaluation will be released publicly upon acceptance.

REFERENCES

Corey Adams, Rishikesh Ranade, Ram Cherukuri, and Sanjay Choudhry. Geotransolver: Learning physics on irregular domains using multi-scale geometry aware physics attention transformer. *arXiv preprint arXiv:2512.20399*, 2025.

Sabine Attinger and Petros Koumoutsakos (eds.). *Multiscale Modelling and Simulation*, volume 39 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, Heidelberg, 2004. doi: 10.1007/978-3-642-18756-8.

Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5):320–328, 2024. doi: 10.1038/s42254-024-00712-5.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Katia Bertoldi, Vincenzo Vitelli, Johan Christensen, and Martin van Hecke. Flexible mechanical metamaterials. *Nature Reviews Materials*, 2(11):17066, 2017. doi: 10.1038/natrevmats.2017.66.

M. A. Bessa, R. Bostanabad, Z. Liu, A. Hu, Daniel W. Apley, C. Brinson, W. Chen, and Wing Kam Liu. A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality. *Computer Methods in Applied Mechanics and Engineering*, 320:633–667, 2017. doi: 10.1016/j.cma.2017.03.037.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations (ICLR)*, 2022.

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24924–24940, 2021.

Nicolas Charalambakis. Homogenization techniques and micromechanics. A survey and perspectives. *Applied Mechanics Reviews*, 63(3):030803, 2010. doi: 10.1115/1.4001911.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd edition, 1988.

Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert J. Witt. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, 4th edition, 2001.

Ojan Khatib Damavandi, Varda F Hagh, Christian D Santangelo, and M Lisa Manning. Energetic rigidity. ii. applications in examples of biological and underconstrained materials. *Physical Review E*, 105(2):025004, 2022a.

Ojan Khatib Damavandi, M Lisa Manning, and JM Schwarz. Effective medium theory of random regular networks. *Europhysics Letters*, 138(2):27001, 2022b.

Ojan Khatib Damavandi, Sadjad Arzash, Elizabeth Lawson-Keister, and M. Lisa Manning. Universality in the mechanical behavior of vertex models for biological tissues. *PRX Life*, 3(3):033001, 2025. doi: 10.1103/9ktk-6rqc.

Jacob Fish and Amir Wagiman. Multiscale finite element method for a locally nonperiodic heterogeneous medium. *Computational Mechanics*, 12(3):164–180, 1993. doi: 10.1007/BF00371991.

Huajian Gao, Baohua Ji, Ingomar L Jäger, Eduard Arzt, and Peter Fratzl. Materials become insensitive to flaws at nanoscale: lessons from nature. *Proceedings of the national Academy of Sciences*, 100(10):5597–5600, 2003.

Marc G. D. Geers, Varvara G. Kouznetsova, and W. A. M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics*, 234(7):2175–2182, 2010. doi: 10.1016/j.cam.2009.08.077.

Marc G. D. Geers, Varvara G. Kouznetsova, Karel Matouš, and Julien Yvonnet. Homogenization methods and multiscale modeling: Nonlinear problems. In *Encyclopedia of Computational Mechanics*, pp. 1–34. John Wiley & Sons, second edition, 2017. doi: 10.1002/9781119176817.ecm107.

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.

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*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12556–12569. PMLR, 2023.

Junyan He, Seid Koric, Shashank Kushwaha, Jaewan Park, Diab Abueidda, and Iwona Jasiuk. Novel deeponet architecture to predict stresses in elastoplastic structures with variable complex geometries and loads. *Computer Methods in Applied Mechanics and Engineering*, 415:116277, 2023.

Junyan He, Shashank Kushwaha, Jaewan Park, Seid Koric, Diab Abueidda, and Iwona Jasiuk. Sequential deep operator networks (S-DeepONet) for predicting full-field solutions under time-dependent loads. *Engineering Applications of Artificial Intelligence*, 127:107258, 2024. doi: 10.1016/j.engappai.2023.107258.

Rodney Hill. Elastic properties of reinforced solids: some theoretical principles. *Journal of the Mechanics and Physics of Solids*, 11(5):357–372, 1963.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, NY, 2000.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality decomposed: How do neural networks generalise? *Journal of Artificial Intelligence Research*, 67:757–795, 2020. doi: 10.1613/jair.1.11674.

George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. doi: 10.1038/s42254-021-00314-5.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations (ICLR)*, 2020.

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. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Manik Kumar, Nilay Upadhyay, Shishir Barai, Wesley F Reinhart, and Christian Peco. A bio-lattice deep learning framework for modeling discrete biological materials. *Journal of the Mechanical Behavior of Biomedical Materials*, 164:106900, 2025.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2873–2882. PMLR, 2018.

Jose Antonio Lara Benitez, Takashi Furuya, Florian Faucher, Anastasis Kratsios, Xavier Tricoche, and Maarten V. de Hoop. Out-of-distributional risk bounds for neural operators with applications to the Helmholtz equation. *Journal of Computational Physics*, 513:113168, 2024. doi: 10.1016/j.jcp.2024.113168.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023a.

Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for PDE surrogate modeling. In *Advances in Neural Information Processing Systems*, volume 36, pp. 28010–28039, 2023b.

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 (ICLR)*, 2021.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 35836–35854, 2023c.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

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):218–229, 2021.

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 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41432–41449. PMLR, 2025.

Grigorios A. Pavliotis and Andrew M. Stuart. *Multiscale Methods: Averaging and Homogenization*, volume 53 of *Texts in Applied Mathematics*. Springer, New York, 2008. doi: 10.1007/978-0-387-73829-1.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi: 10.1016/j.jcp.2018.10.045.

Junuthula Narasimha Reddy. *An Introduction to the Finite Element Method*. McGraw-Hill, New York, 2nd edition, 1993.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 2020.

Robert Schittny, Muamer Kadic, Sebastien Guenneau, and Martin Wegener. Experiments on transformation thermodynamics: Molding the flow of heat. *Physical Review Letters*, 110(19):195901, 2013. doi: 10.1103/PhysRevLett.110.195901.

Harsh Sharma, Ajay Kumar, Sravendra Rana, Nanda Gopal Sahoo, Muhammad Jamil, Rajeev Kumar, Shubham Sharma, Changhe Li, Abhinav Kumar, Sayed M Eldin, et al. Critical review on advancements on the fiber-reinforced composites: role of fiber/matrix modification on the performance of the fibrous composites. *Journal of Materials Research and Technology*, 26:2975–3002, 2023.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel 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)*, volume 35, pp. 1596–1611, 2022.

Kenjiro Terada, Muneo Hori, Takashi Kyoya, and Noboru Kikuchi. Simulation of the multi-scale convergence in computational homogenization approaches. *International Journal of Solids and Structures*, 37(16):2285–2311, 2000.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized Fourier neural operators. In *International Conference on Learning Representations (ICLR)*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705. PMLR, 2024.

Tianju Xue, Shuheng Liao, Zhengtao Gan, Chanwook Park, Xiaoyu Xie, Wing Kam Liu, and Jian Cao. JAX-FEM: A differentiable GPU-accelerated 3D finite element solver for automatic inverse design and mechanistic data science. *Computer Physics Communications*, 291:108802, 2023. doi: 10.1016/j.cpc.2023.108802.

Zhiyuan Zhao, Xueying Ding, and B Aditya Prakash. PINNsFormer: A transformer-based framework for physics-informed neural networks. In *International Conference on Learning Representations (ICLR)*, 2024.

Hang Zhou, Haixu Wu, Haonan Shangguan, Yuezhou Ma, Huikun Weng, Jianmin Wang, and Mingsheng Long. Transolver-3: Scaling up transformer solvers to industrial-scale geometries. *arXiv preprint arXiv:2602.04940*, 2026.

Jiayi Zhou, Valentin Duruisseaux, Daniel Zhengyu Huang, and Anima Anandkumar. Boundary-augmented neural operators for better generalization to unseen geometries. In *NeurIPS 2025 Workshop on AI for Science*, 2025. URL https://openreview.net/forum?id=DqZoWaDwfN.

A Finite Element Formulation

This appendix outlines the finite element formulation used to generate the dataset for the heterogeneous elasticity PDE. Let $\Omega \subset \mathbb{R}^2$ denote the computational domain, and let the displacement field be

$$\mathbf{u}(\mathbf{x}) = [u_x(\mathbf{x}), u_y(\mathbf{x})]^T. \tag{12}$$

The domain boundary is decomposed into Dirichlet and Neumann parts, denoted by Γ_g and Γ_h, respectively. The strong form of linear momentum balance under quasistatic conditions is

$$\begin{aligned} \sigma_{ij,j} + l_i &= 0, && \text{in } \Omega, \\ u_i &= g_i, && \text{on } \Gamma_g, \\ \sigma_{ij} n_j &= h_i, && \text{on } \Gamma_h, \end{aligned} \tag{13}$$

where σ_{ij} is the Cauchy stress tensor, l_i is the body force, g_i is the prescribed displacement on the Dirichlet boundary, h_i is the prescribed traction on the Neumann boundary, and n_j is the outward unit normal. In the present simulations, body forces and surface tractions are neglected:

$$l_i = 0, \qquad h_i = 0. \tag{14}$$

The left boundary is constrained by $u_x = 0$, and $u_y = 0$ is imposed at the bottom-left corner to remove rigid-body motion. A horizontal displacement $u_x = \Delta$ is prescribed on the right boundary, where $\Delta \sim \mathcal{U}(0.02, 0.08)$ is sampled independently for each case. We obtain the weak form by multiplying the equilibrium equation by an admissible test function w_i and integrating over the domain. The trial and test spaces are defined as

$$\mathcal{U} = \left\{ \mathbf{u} \in [H^1(\Omega)]^2 \mid \mathbf{u} = \mathbf{g} \text{ on } \Gamma_g \right\}, \tag{15}$$

and

$$\mathcal{V} = \left\{ \mathbf{w} \in [H^1(\Omega)]^2 \mid \mathbf{w} = \mathbf{0} \text{ on } \Gamma_g \right\}. \tag{16}$$

The weak form seeks $\mathbf{u} \in \mathcal{U}$ such that, for all $\mathbf{w} \in \mathcal{V}$,

$$\int_\Omega \varepsilon_{ij}(\mathbf{w}) \sigma_{ij}(\mathbf{u}) \, d\Omega = 0. \tag{17}$$

Here, the infinitesimal strain tensor is

$$\varepsilon_{ij}(\mathbf{u}) = u_{(i,j)} = \frac{1}{2} \left(u_{i,j} + u_{j,i} \right). \tag{18}$$

For linear elasticity, the constitutive relation is

$$\sigma_{ij} = c^{(e)}_{ijkl} \varepsilon_{kl}, \tag{19}$$

where $c^{(e)}_{ijkl}$ is the elasticity tensor of element e. The superscript indicates that material properties may vary from element to element because of the heterogeneous matrix–inclusion structure.

Using finite-dimensional approximation spaces, the displacement field is approximated as

$$\mathbf{u}^h(\mathbf{x}) = \sum_{a=1}^{n_n} N_a(\mathbf{x}) \mathbf{d}_a, \tag{20}$$

where N_a is the shape function associated with node a, $\mathbf{d}_a = [d_{x,a}, d_{y,a}]^T$ is the nodal displacement vector, and n_n is the number of nodes per element. This work uses three-node triangular elements. The shape functions for a linear triangular element are

$$N_a(x, y) = \frac{1}{2A} \left(a_a + b_a x + c_a y \right), \qquad a = 1, 2, 3, \tag{21}$$

where the element area is

$$A = \frac{1}{2}\left[x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)\right]. \tag{22}$$

The coefficients are

$$a_a = x_b y_c - x_c y_b, \qquad b_a = y_b - y_c, \qquad c_a = x_c - x_b, \tag{23}$$

where (a, b, c) follows cyclic permutation over the three element nodes. The derivatives of the shape functions are constant within each element:

$$\frac{\partial N_a}{\partial x} = \frac{b_a}{2A}, \qquad \frac{\partial N_a}{\partial y} = \frac{c_a}{2A}. \tag{24}$$

For two-dimensional linear elasticity, the element strain vector is written in Voigt notation as

$$\boldsymbol{\varepsilon}^{(e)} = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{bmatrix} = \mathbf{B}^{(e)}\mathbf{d}^{(e)}, \tag{25}$$

where $\gamma_{xy} = 2\varepsilon_{xy}$ is the engineering shear strain and $\mathbf{d}^{(e)}$ is the element nodal displacement vector. For a three-node triangular element,

$$\mathbf{B}^{(e)} = \frac{1}{2A}\begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & c_2 & 0 & c_3 \\ c_1 & b_1 & c_2 & b_2 & c_3 & b_3 \end{bmatrix}. \tag{26}$$

For plane stress elasticity, the material matrix is

$$\mathbf{D}^{(e)} = \frac{E^{(e)}}{1 - \nu^{(e)2}}\begin{bmatrix} 1 & \nu^{(e)} & 0 \\ \nu^{(e)} & 1 & 0 \\ 0 & 0 & \frac{1-\nu^{(e)}}{2} \end{bmatrix}, \tag{27}$$

where $E^{(e)}$ and $\nu^{(e)}$ are the Young's modulus and Poisson's ratio of element e. For heterogeneous microstructures, assign these material parameters based on whether the element belongs to the matrix or inclusion phase. The element stiffness matrix is

$$\mathbf{k}^{(e)} = \int_{\Omega_e} \mathbf{B}^{(e)T}\mathbf{D}^{(e)}\mathbf{B}^{(e)}\, d\Omega. \tag{28}$$

Since the strain–displacement matrix is constant for a linear triangular element, this becomes

$$\mathbf{k}^{(e)} = A_e \mathbf{B}^{(e)T}\mathbf{D}^{(e)}\mathbf{B}^{(e)}, \tag{29}$$

where unit thickness is assumed. The global stiffness matrix is assembled from all element contributions:

$$\mathbf{K} = \sum_{e=1}^{n_e} \mathbf{A}^{(e)T}\mathbf{k}^{(e)}\mathbf{A}^{(e)}, \tag{30}$$

where $\mathbf{A}^{(e)}$ is the element-to-global assembly operator.

After applying the prescribed displacement boundary conditions, the discrete finite element system can be written as

$$\mathbf{K}\mathbf{d} = \mathbf{F}, \tag{31}$$

where $\mathbf{K}$ is the assembled stiffness matrix, $\mathbf{d}$ is the vector of unknown nodal displacements, and $\mathbf{F}$ contains the effective force contribution resulting from the imposed boundary conditions. The solved nodal displacement field is used as the reference output for training and evaluating the neural PDE surrogates. We generate finite element meshes using *Gmsh* and solve the corresponding linear-elasticity problems using *JAX-FEM*. Each dataset sample contains the nodal coordinates, element connectivity, material descriptors, boundary-condition descriptors, and reference nodal displacement field. We then construct the material masks and continuous material-influence descriptors used by CCR-MIG from the material distribution, as described in Section 4.2.

B STEADY THERMAL-CONDUCTION FINITE ELEMENT FORMULATION

This appendix summarizes the FEA formulation for the heterogeneous thermal-conduction PDE benchmark. Let

$$\Omega = [0, 10] \times [0, 5] \subset \mathbb{R}^2 \tag{32}$$

denote the computational domain, and let

$$T(\mathbf{x}) \tag{33}$$

denote the steady temperature field. The thermal conductivity $k(\mathbf{x})$ is spatially heterogeneous and piecewise constant within each finite element according to the matrix or inclusion phase. The steady heat-conduction equation in the absence of volumetric heat generation is

$$-\nabla \cdot (k(\mathbf{x})\nabla T) = 0, \qquad \mathbf{x} \in \Omega. \tag{34}$$

Fourier's law gives the corresponding conductive heat-flux vector,

$$\mathbf{q} = -k(\mathbf{x})\nabla T. \tag{35}$$

The left and right boundaries are subjected to prescribed temperatures,

$$T = T_{\text{hot}} = 1 \qquad \text{on } x = 0, \tag{36}$$

and

$$T = T_{\text{cold}} = 0 \qquad \text{on } x = 10. \tag{37}$$

The upper and lower boundaries are thermally insulated, such that

$$\mathbf{q} \cdot \mathbf{n} = 0 \qquad \text{on } y = 0 \text{ and } y = 5, \tag{38}$$

where $\mathbf{n}$ denotes the outward unit normal.

Let

$$\Gamma_T = \Gamma_{\text{hot}} \cup \Gamma_{\text{cold}} \tag{39}$$

denote the Dirichlet portion of the boundary and Γ_q denotes the insulated portion. The trial space is

$$\mathcal{U}_T = \left\{ T \in H^1(\Omega) \mid T = T_{\text{hot}} \text{ on } \Gamma_{\text{hot}},\ T = T_{\text{cold}} \text{ on } \Gamma_{\text{cold}} \right\}, \tag{40}$$

and the corresponding test space is

$$\mathcal{V}_T = \left\{ w \in H^1(\Omega) \mid w = 0 \text{ on } \Gamma_T \right\}. \tag{41}$$

Multiplying Eq. (34) by an admissible test function w and integrating by parts gives

$$\int_\Omega k(\mathbf{x})\nabla w \cdot \nabla T \, d\Omega - \int_{\Gamma_q} w \, \mathbf{n} \cdot k\nabla T \, d\Gamma = 0. \tag{42}$$

Because the top and bottom boundaries are insulated, the boundary integral vanishes, yielding the weak form: find $T \in \mathcal{U}_T$ such that

$$\int_\Omega k(\mathbf{x})\nabla w \cdot \nabla T \, d\Omega = 0 \qquad \forall w \in \mathcal{V}_T. \tag{43}$$

The temperature field is approximated using three-node linear triangular elements:

$$T^h(\mathbf{x}) = \sum_{a=1}^{3} N_a(\mathbf{x}) T_a, \tag{44}$$

where N_a are the same linear triangular shape functions defined in Appendix A and T_a denotes the nodal temperature. The temperature gradient within element e is therefore

$$\nabla T^{(e)} = \mathbf{B}_T^{(e)} \mathbf{T}^{(e)}, \tag{45}$$

where

$$\mathbf{T}^{(e)} = \begin{bmatrix} T_1 & T_2 & T_3 \end{bmatrix}^T, \tag{46}$$

and

$$\mathbf{B}_T^{(e)} = \begin{bmatrix} \partial N_1/\partial x & \partial N_2/\partial x & \partial N_3/\partial x \\ \partial N_1/\partial y & \partial N_2/\partial y & \partial N_3/\partial y \end{bmatrix}. \tag{47}$$

For a piecewise-constant element conductivity $k^{(e)}$, the elemental conductivity matrix is

$$\mathbf{K}_T^{(e)} = \int_{\Omega_e} \mathbf{B}_T^{(e)T} k^{(e)} \mathbf{B}_T^{(e)} \, d\Omega. \tag{48}$$

Because $\mathbf{B}_T^{(e)}$ and $k^{(e)}$ are constant within a linear triangular element,

$$\mathbf{K}_T^{(e)} = A_e \, \mathbf{B}_T^{(e)T} k^{(e)} \mathbf{B}_T^{(e)}, \tag{49}$$

where unit out-of-plane thickness is assumed.

The global conductivity matrix is assembled as

$$\mathbf{K}_T = \sum_{e=1}^{n_e} \mathbf{A}^{(e)T} \mathbf{K}_T^{(e)} \mathbf{A}^{(e)}, \tag{50}$$

where $\mathbf{A}^{(e)}$ denotes the element-to-global assembly operator. After applying the prescribed temperatures, the discrete system is written as

$$\mathbf{K}_T \mathbf{T} = \mathbf{F}_T, \tag{51}$$

where $\mathbf{T}$ contains the unknown nodal temperatures and $\mathbf{F}_T$ contains the effective contributions arising from the imposed Dirichlet temperatures.

The element heat flux is recovered from

$$\mathbf{q}^{(e)} = -k^{(e)} \mathbf{B}_T^{(e)} \mathbf{T}^{(e)}. \tag{52}$$

For the thermal benchmark, the matrix conductivity is set to

$$k_m = 1. \tag{53}$$

To preserve the dimensionless constituent contrast used in the elasticity benchmark, the non-mixed thermal training data use

$$\frac{k^{(e)}}{k_m} = \frac{E^{(e)}}{E_m}. \tag{54}$$

Thus,

$$\frac{k_{\text{low}}}{k_m} \in [0.05, 0.5], \qquad \frac{k_{\text{high}}}{k_m} \in [2, 20]. \tag{55}$$

The effective conductivity in the imposed horizontal transport direction can also be computed from the net outlet heat rate as

$$k_{\text{eff}} = \frac{Q_{\text{out}} L_x}{L_y \left(T_{\text{hot}} - T_{\text{cold}}\right)}, \tag{56}$$

for unit out-of-plane thickness. This quantity is used only as a numerical consistency diagnostic and is not a learning target. The thermal finite element problems are solved using JAX-FEM on conforming unstructured triangular meshes generated using *Gmsh*. The solved nodal temperature field is used as the scalar reference output for training and evaluating the neural PDE surrogates. Each thermal dataset sample contains the nodal coordinates, element connectivity, conductivity and material descriptors, boundary-condition descriptors, and the reference temperature field.

C Physics-Attention Background

This appendix briefly summarizes the original Physics-Attention mechanism proposed in Transolver, which serves as the backbone of the present work. The complete derivation can be found in the original Transolver publications. Here we include only the mathematical formulation necessary to understand the proposed MIG extensions presented in Section 4.

C.1 PHYSICS-ATTENTION

Unlike conventional Transformers that perform self-attention directly among mesh nodes, Transolver first compresses the mesh into a small number of latent physical states (also referred to as slices), performs attention among these states, and finally broadcasts the updated information back to individual mesh nodes.
Given an input mesh consisting of N nodes,

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times d_{\text{in}}}, \tag{57}$$

each node contains its geometric coordinates and physical descriptors. A linear embedding projects the input into a latent feature space,

$$\mathbf{X}_p = \text{Linear}(\mathbf{X}). \tag{58}$$

These latent features are subsequently grouped into a fixed number of physics-aware slices.

C.2 SLICE OPERATION

Instead of treating every mesh node as an individual attention token, Transolver learns a soft assignment from mesh nodes to latent slices. Slice weights are computed as

$$\mathbf{W} = \text{Softmax}\left(\text{Linear}(\mathbf{X})\right), \qquad \mathbf{W} \in \mathbb{R}^{N \times K}, \tag{59}$$

where K denotes the number of latent slices. Each element w_{ij} represents the contribution of mesh node i to slice j. The latent slice representations are computed by normalized weighted aggregation,

$$\mathbf{S} = (\mathbf{W}\mathbf{D}^{-1})^T \mathbf{X}_p, \tag{60}$$

where

$$D_{jj} = \sum_{i=1}^{N} w_{ij} \tag{61}$$

is the normalization factor for slice j. The slice operation reduces the original mesh representation from N mesh nodes to only K latent physical states.

C.3 ATTENTION AMONG SLICE TOKENS

Self-attention is performed only among the latent slice tokens rather than directly among mesh nodes. The query, key and value matrices are computed as

$$\mathbf{Q} = \mathbf{S}\mathbf{W}_Q, \qquad \mathbf{K} = \mathbf{S}\mathbf{W}_K, \qquad \mathbf{V} = \mathbf{S}\mathbf{W}_V. \tag{62}$$

The updated slice representations are obtained through scaled dot-product attention,

$$\mathbf{S}' = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{63}$$

where d denotes the hidden feature dimension.

C.4 DESLICE OPERATION

After attention among latent slices, the updated information is broadcast back to individual mesh nodes using the same slice weights. The deslice operation is written as

$$\mathbf{X}' = \mathbf{W}\mathbf{S}'. \tag{64}$$

Finally,

$$\hat{\mathbf{X}} = \text{Linear}(\mathbf{X}') \tag{65}$$

produces the output features of the Physics-Attention block. The complete block is embedded within a standard Transformer layer using residual connections, layer normalization and a feed-forward network.

C.5 Computational Complexity

Physics-Attention performs self-attention among K latent slice tokens rather than directly among all N mesh nodes. Consequently, the slice-space attention operation has complexity

$$\mathcal{O}(K^2), \tag{66}$$

whereas conventional node-wise self-attention requires

$$\mathcal{O}(N^2). \tag{67}$$

Accounting also for the node-to-slice aggregation and deslice operations, the dominant complexity of the Physics-Attention representation can be expressed approximately as

$$\mathcal{O}(NK + K^2), \tag{68}$$

with $K \ll N$. This latent-state representation therefore enables long-range information exchange while avoiding quadratic attention over all mesh nodes.

C.6 Difference Between Transolver and CCR-MIG

CCR-MIG preserves the overall Physics-Attention architecture of Transolver. The slice-space self-attention, deslice operation, residual connections, feed-forward network, and output prediction head remain unchanged. The proposed modifications concern the information supplied to the model and the construction of the latent slice assignments. The distinction can be separated into two pathways.

Content conditioning The original Transolver receives the baseline nodal descriptors $\mathbf{f}_i$. CCR-MIG augments this representation with continuous soft- and hard-material influence descriptors,

$$\widetilde{\mathbf{f}}_i = \left[\mathbf{f}_i,\, a_i^s,\, a_i^h,\, a_i^s + a_i^h\right]. \tag{69}$$

A feature-matched Transolver baseline receives the same augmented representation while retaining the original Physics-Attention routing. This baseline isolates the effect of material information supplied only through the input features.

Routing conditioning For the original Transolver, the slice assignment distribution of node i is determined entirely from the learned routing logits,

$$\mathbf{p}_i^{\text{free}} = \operatorname{softmax}\left(\frac{\mathbf{s}_i}{\tau}\right). \tag{70}$$

CCR-MIG additionally constructs a material-conditioned routing distribution $\mathbf{q}_i$ from constituent-specific slice distributions. Rather than adding a material bias directly to the routing logits, CCR forms the final slice assignment through the convex mixture

$$\mathbf{p}_i = (1 - \gamma_i)\mathbf{p}_i^{\text{free}} + \gamma_i \mathbf{q}_i, \qquad 0 \le \gamma_i \le \gamma_{\max}. \tag{71}$$

A conflict-aware gating mechanism reduces γ_i when soft and hard material influences coexist. This behavior is designed for the compositional-generalization setting considered in this work, where the individual constituents are observed separately during training but their simultaneous occurrence is unseen. Because both $\mathbf{p}_i^{\text{free}}$ and $\mathbf{q}_i$ are probability distributions, the routing intervention is bounded by

$$\left\|\mathbf{p}_i - \mathbf{p}_i^{\text{free}}\right\|_1 \le 2\gamma_i \le 2\gamma_{\max}. \tag{72}$$

Therefore, CCR-MIG differs from the original Transolver not by replacing Physics-Attention, but by introducing material information through both the nodal feature representation and a bounded modification of latent-state formation. The complete CCR formulation is given in Section 4.3.

D Additional Qualitative Results and Latent-State Analysis

This appendix provides additional qualitative results for zero-shot mixed composition and geometric distribution shift, followed by visualizations of the learned latent routing and slice-space attention. For each PDE, predictions are obtained using the corresponding trained CCR-MIG model.

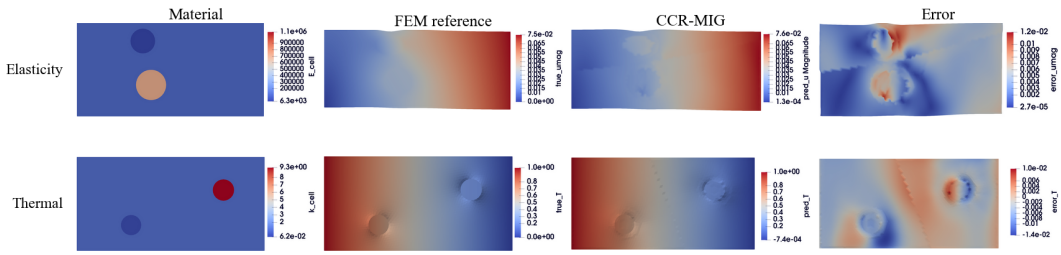

Figure 6: Zero-shot mixed-composition test samples. The elasticity example contains one soft and one hard inclusion, while the thermal example contains one low- and one high-conductivity inclusion. From left to right: material distribution, finite element reference, CCR-MIG prediction, and pointwise error.

Zero-shot mixed composition We examine the mixed composition test set, which consists of 200 samples. Figure 6 shows representative examples. We compare the CCR-MIG displacement and temperature predictions with the corresponding finite element reference solutions. The predicted fields capture the overall features of the reference solutions. The main differences are in material interfaces and inclusions.

Geometric distribution shift We also evaluate the trained CCR-MIG model on inclusion shapes that are not present in the circular-inclusion training data. These examples include an ellipse and randomly generated smooth blobs with soft or hard material properties. Figure 7 compares the predicted fields with the finite element reference solutions. These examples show how the trained model responds when the inclusion geometry changes from the circular shapes used during training. Quantitative comparison with the baseline models under the same geometric shift is left for future work.

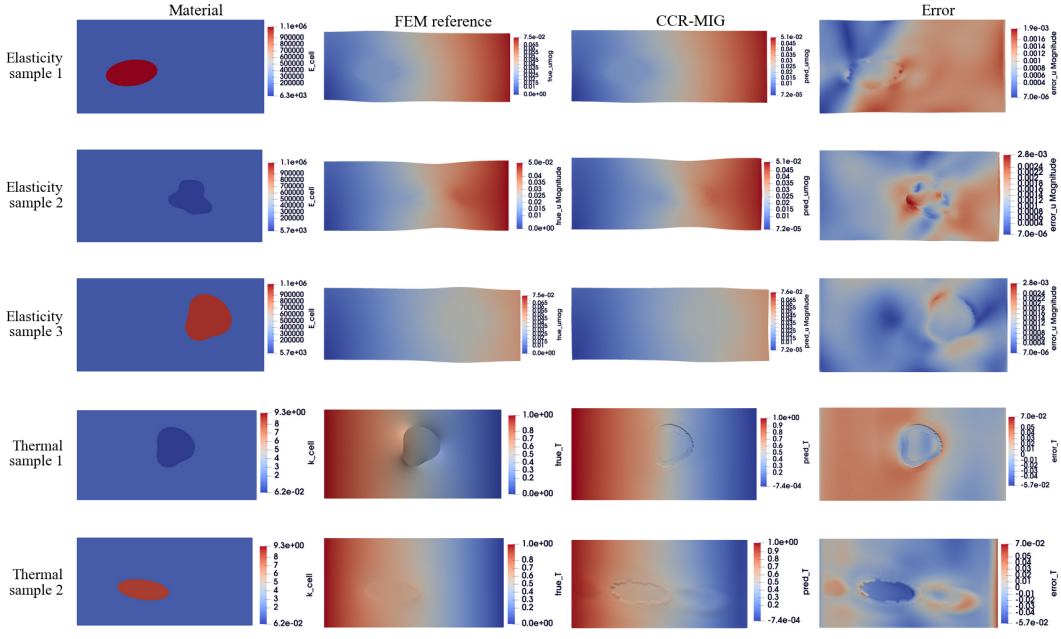

Figure 7: Geometric distribution shift evaluated without retraining. The examples include ellipses and smooth blobs with soft or hard elastic properties or low or high thermal conductivity, none of which appear in the circular-inclusion training distribution. From left to right: material distribution, finite element reference, CCR-MIG prediction, and pointwise error.

D.1 Learned Latent Slices and Slice-Space Attention

We also look at how nodes are assigned to latent slices and how the slices interact with each other. All results are from the same trained elasticity model, with no additional training.

Figure 8 compares the original Physics-Attention routing p_i^{free}, the material-guided distribution q_i, and the final CCR-MIG routing p_i. The learned centers are approximately 5.70, 17.46, 29.96, and 43.15 for the soft, hard, soft-interface, and hard-interface categories, respectively. After adding the material-guided routing, the final routing is still close to the original Transolver routing.

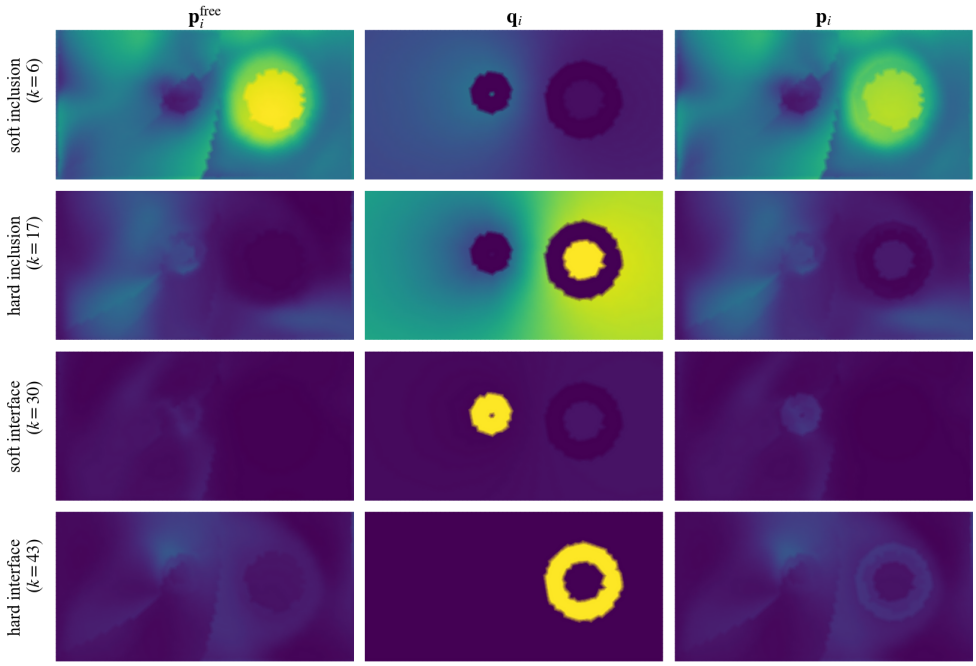

Figure 8: Routing comparison at four learned category-centered slices for a representative zero-shot mixed-composition sample. Columns show the original Physics-Attention routing p_i^{free}, the material-guided distribution q_i, and the final CCR-MIG routing p_i. Rows correspond to the soft, hard, soft-interface, and hard-interface routing categories.

We also compare the learned routing after randomly retaining 70% of the original nodes while preserving the boundary nodes. Figure 9 shows the original and subsampled cases. After keeping 70% of the original nodes, the dominant slice assignment is the same for 97.0% of the retained nodes. The mean nodewise routing ℓ_1 difference is 0.0178, and the relative L_2 difference between the two predictions is 0.00577. This shows that the routing changes only slightly after node subsampling in this example.

Figure 10 shows the self-attention between the 48 latent slice tokens in the same Physics-Attention layer. The attention values are not uniformly distributed across the slice pairs. Under the same node-subsampling test, the mean absolute difference in the head-averaged attention matrix is 1.89×10^{-3}.

E Additional Implementation and Experimental Details

E.1 CCR Routing-Gate Parameterization

The routing intervention in CCR is controlled independently within each Physics-Attention layer and attention head. For layer ℓ and head h, we introduce a learnable scalar $g_{\ell,h}$ and define the bounded base routing strength as

$$\bar{\gamma}_{\ell,h} = \gamma_{\max}\sigma(g_{\ell,h}), \tag{73}$$

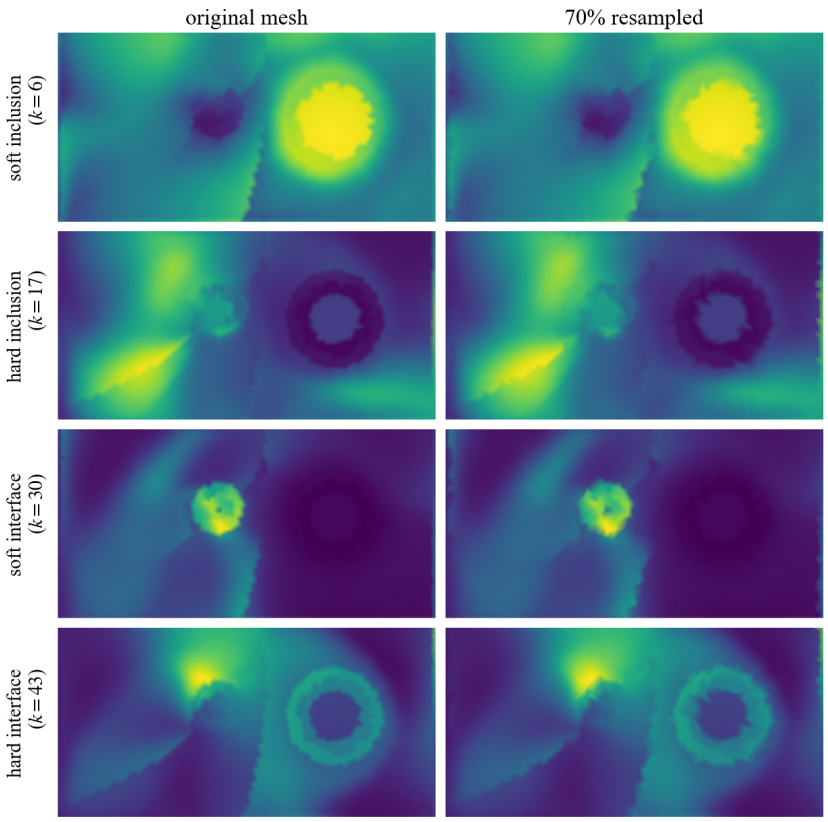

Figure 9: Learned slice assignments for the original node set (left) and a random 70% node subsample (right) of the same physical configuration. The same trained CCR-MIG model is used in both cases. Rows correspond to the learned soft, hard, soft-interface, and hard-interface category-centered slices.

where $\sigma(\cdot)$ is the sigmoid function and $\gamma_{\max} = 0.30$. The gate is therefore not predicted independently at every node by an MLP. Instead, each attention head in each layer learns its own base routing strength.

Nodewise variation is introduced through the material-routing activity A_i and the constituent-conflict measure C_i:

$$\gamma_{i,\ell,h} = \bar{\gamma}_{\ell,h} A_i \left(1 - \lambda_C C_i\right), \tag{74}$$

where $\lambda_C = 0.75$. The resulting gate is clipped to $[0, \gamma_{\max}]$.

The parameter `route_gate_init`=0.05 refers to the initial *effective* value of the routing gate rather than to the raw learnable parameter. The initialization is chosen such that

$$\gamma_{\max} \sigma(g_{\ell,h}^{(0)}) = 0.05 \tag{75}$$

for every head. Thus, CCR begins as a relatively weak intervention to the original Physics-Attention routing and its strength is learned during training.

E.2 Continuous Constituent-Influence Fields

The content-conditioning pathway uses two continuous constituent-influence fields, one for the soft constituent and one for the hard constituent. For $c \in \{s, h\}$,

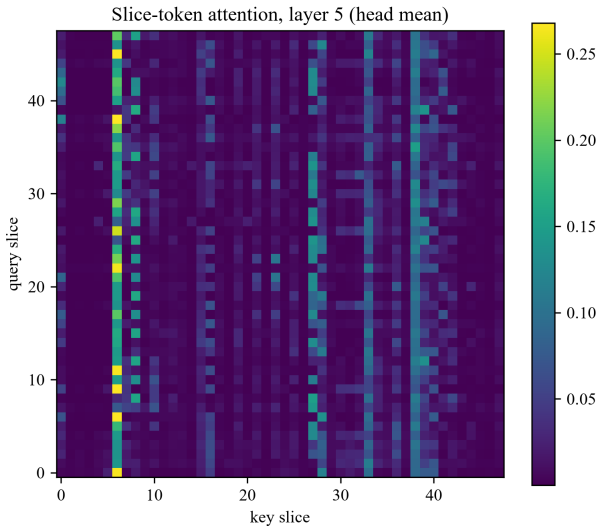

Figure 10: Head-averaged self-attention between the 48 latent slice tokens in the final Physics-Attention layer for a representative mixed-composition sample. Rows denote query slices and columns denote key slices.

$$a_i^{(c)} = \left(\frac{r_c}{r_c + \eta d_{i,c}} \right)^2, \qquad \eta = 0.6, \tag{76}$$

followed by clipping to $[0, 1]$. Here $d_{i,c}$ is the minimum Euclidean distance from node i to the support of constituent c, while r_c is estimated from the spatial area occupied by the corresponding constituent. A minimum numerical radius of 10^{-3} is imposed.

For the soft influence field, the source support includes both soft-inclusion and soft-interface nodes. Likewise, the hard field uses the union of hard-inclusion and hard-interface nodes.

The continuous descriptors appended to the feature vector in the reported configuration are

$$\mathbf{a}_i = \left[a_i^s,\ a_i^h,\ a_i^s + a_i^h \right]. \tag{77}$$

The interaction product $a_i^s a_i^h$ is disabled. Therefore, the model does not use four separate continuous inclusion/interface influence fields. Instead, the four-way inclusion/interface distinction is retained in the routing pathway described below.

E.3 Material and Interface Routing Categories

CCR uses four routing categories,

$$\mathcal{C}_{\text{route}} = \{\text{soft-inclusion}, \text{hard-inclusion}, \text{soft-interface}, \text{hard-interface}\}. \tag{78}$$

Each category c is associated with a learnable Gaussian distribution over the K latent slices:

$$q_c(k) = \frac{\exp\left[-\dfrac{(k - \mu_c)^2}{2\sigma_c^2 \tau_m} \right]}{\sum_{j=1}^{K} \exp\left[-\dfrac{(j - \mu_c)^2}{2\sigma_c^2 \tau_m} \right]}, \tag{79}$$

where μ_c and σ_c are learnable and $\tau_m = 1.0$.

For $K = 48$ slices, the four Gaussian centers are initialized uniformly as

$$\mu_c^{(0)} \in \{6, 18, 30, 42\}, \tag{80}$$

and their initial widths are

$$\sigma_c^{(0)} = 0.12K = 5.76. \tag{81}$$

The width is optimized in log space to maintain positivity.

Let $a_{i,c}$ denote the routing activity associated with category c. The normalized category shares are

$$\pi_{i,c} = \frac{a_{i,c}}{\sum_{c'} a_{i,c'} + \epsilon}, \tag{82}$$

and the node-specific material routing distribution is

$$q_i = \sum_c \pi_{i,c} q_c. \tag{83}$$

E.4 Constituent Conflict

The soft and hard constituent masses used in the conflict measure combine their respective inclusion and interface categories:

$$\pi_{i,s} = \pi_{i,\mathrm{soft-inc}} + \pi_{i,\mathrm{soft-int}}, \tag{84}$$

$$\pi_{i,h} = \pi_{i,\mathrm{hard-inc}} + \pi_{i,\mathrm{hard-int}}. \tag{85}$$

The conflict measure is

$$C_i = 4\pi_{i,s}\pi_{i,h}, \qquad C_i \in [0, 1]. \tag{86}$$

It approaches zero when one constituent dominates and reaches its maximum when the normalized soft and hard contributions are balanced.

For each head, the final routing distribution is

$$p_{i,\ell,h} = (1 - \gamma_{i,\ell,h})\, p_{i,\ell,h}^{\mathrm{free}} + \gamma_{i,\ell,h} q_i. \tag{87}$$

Since both terms are probability distributions and $\gamma_{i,\ell,h} \le \gamma_{\max}$,

$$\left\| p_{i,\ell,h} - p_{i,\ell,h}^{\mathrm{free}} \right\|_1 \le 2\gamma_{i,\ell,h} \le 2\gamma_{\max}. \tag{88}$$

E.5 Training Configuration

All compared models use six Physics-Attention layers, a hidden dimension of 192, eight attention heads, and 48 latent slices. Models are optimized for 1000 epochs using AdamW with an initial learning rate of 10^{-3}, weight decay 10^{-5}, and maximum gradient norm 1.0. A cosine-annealing learning-rate schedule is applied over the full training horizon without warmup.

Because the independently generated conforming meshes contain different numbers of nodes, both training and validation use a batch size of one.

The main CCR hyperparameters are

$$\gamma_{\max} = 0.30, \qquad \gamma_{\mathrm{init}} = 0.05, \qquad \lambda_C = 0.75, \qquad \tau_m = 1.0. \tag{89}$$

E.6 Dataset and Inclusion Statistics

The non-mixed training benchmark contains four classes: single-soft, single-hard, multi-soft, and multi-hard. Single-inclusion samples contain one inclusion, whereas multi-inclusion samples contain between two and three inclusions of the same constituent class. The training radius range is

$$R_{\text{train}} \in [0.35, 0.90]. \tag{90}$$

The mixed elasticity benchmark contains exactly one soft and one hard inclusion per sample. The corresponding thermal benchmark contains one low-conductivity and one high-conductivity inclusion per sample. The mixed generators use

$$R_{\text{mixed}} \in [0.35, 1.00]. \tag{91}$$

Thus, all constituent classes and material-property ranges are represented during training, whereas soft–hard or low–high co-occurrence is withheld. The inclusion-count and radius distributions are not strictly identical between the non-mixed training and mixed evaluation sets; therefore, we do not claim that constituent co-occurrence is literally the only geometric distribution shift. The training dataset contains 400 samples, of which 360 are used for optimization and 40 for validation. The conforming meshes contain

$$N_{\text{nodes}} \in [2754, 2900], \qquad \overline{N}_{\text{nodes}} = 2813.1, \tag{92}$$

and

$$N_{\text{elem}} \in [5304, 5596], \qquad \overline{N}_{\text{elem}} = 5422.2. \tag{93}$$

F Seed-Level Results and Statistical Analysis

F.1 Seed-Level Results

We train each model with multiple random seeds ($\{123, 456, 789, 1011, 2022, 3033\}$) to account for variability from model initialization and optimization for both the heterogeneous elasticity and thermal-conduction PDEs. For each seed, we calculate the mean relative L_2 error over the same 200 held-out mixed-composition samples. Table 3 and Table 4 show the results for each trained model for elasticity and heat conduction, respectively.

Table 3: Seed-wise relative L_2 errors on the 200-sample mixed-composition elasticity test set. SD is the sample standard deviation across the six runs.

Seed	Transolver	Feature-matched	CCR-MIG
123	0.07563	0.09481	0.04720
456	0.09587	0.06707	0.07926
789	0.08218	0.04901	0.04807
1011	0.07083	0.06261	0.06410
2022	0.03513	0.06196	0.05593
3033	0.06840	0.12572	0.05889
Mean	0.07134	0.07686	**0.05891**
SD	0.02029	0.02831	0.01187

The mean error for Transolver elasticity is 0.07134. For the feature-matched model and CCR-MIG, it is 0.07686 and 0.05891, respectively. CCR-MIG outperforms both Transolver and the feature-matched model by 17.43% and 23.36%, respectively. Notably, the feature-matched model performed worse than the original Transolver in this case.

Table 4: Seed-wise relative L_2 errors on the 200-sample mixed-composition thermal-conduction test set. SD is the sample standard deviation across the six runs.

Seed	Transolver	Feature-matched	CCR-MIG
123	0.02121	0.03170	0.02101
456	0.07328	0.01906	0.02592
789	0.01877	0.03891	0.02720
1011	0.04200	0.03295	0.01739
2022	0.02629	0.02248	0.02368
3033	0.02808	0.02679	0.02423
Mean	0.03494	0.02865	**0.02324**
SD	0.02045	0.00730	0.00356

The mean error for Transolver thermal conduction is 0.03494. For the feature-matched model and CCR-MIG, it is 0.02865 and 0.02324, respectively. CCR-MIG outperforms both Transolver and the feature-matched model by 33.49% and 18.89%, respectively. In this case, the feature-matched model also performs better than the original Transolver.

F.2 Paired Statistical Evaluation

We also compare the trained seed models statistically. For each seed, we first average the relative L_2 error over the 200 held-out samples. We then compare the six paired mean errors using an exact two-sided sign-flip test and paired Cohen's d_z.

Table 5: Paired statistical comparisons on the 200-sample mixed-composition test sets.

Comparison	d_z	Raw p	Holm p
Elasticity: CCR-MIG vs. Transolver	0.6397	0.18750	0.75000
Elasticity: CCR-MIG vs. Feature-matched	0.5686	0.34375	0.75000
Thermal: CCR-MIG vs. Transolver	0.5684	0.28125	0.75000
Thermal: CCR-MIG vs. Feature-matched	0.6265	0.18750	0.75000

The effect sizes are positive for all four comparisons. However, none of the comparisons is statistically significant after Holm correction. We therefore report the mean errors, across-seed variation, and statistical results without making a significance claim.

G Six-Seed Ablation Study

We repeated the ablation study on the 100-sample mixed composition development set using the same six seeds as in Appendix F. We evaluated the following variants:

- **Base**: original Transolver without material feature augmentation or CCR routing;
- **Feature**: material/content feature augmentation without CCR routing;
- **Direct**: unconstrained direct material-dependent routing;
- **CCR-NoConflict**: bounded conservative routing without conflict suppression;
- **Full CCR**: the complete CCR-MIG formulation.

G.1 Elasticity Ablation

Table 6 shows that Full CCR has the lowest mean error, 0.05454. The feature-matched and Direct variants give mean errors of 0.07355 and 0.08700, respectively. Compared with these two variants, Full CCR reduces the mean error by 25.84% and 37.31%, respectively. CCR-NoConflict gives a mean error of 0.05863, which is reduced to 0.05454 with Full CCR. The corresponding paired statistics are reported in Table 7. For elasticity, adding the conflict gate reduces the mean error

Table 6: Seed-wise elasticity DEV-100 ablation results.

Seed	Base	Feature	Direct	NoConflict	Full CCR
123	0.07864	0.08305	0.12930	0.05656	0.04578
456	0.09593	0.06295	0.11694	0.04754	0.06804
789	0.07621	0.04467	0.06129	0.05552	0.04678
1011	0.06421	0.05990	0.05235	0.07948	0.05911
2022	0.03380	0.05851	0.09635	0.06230	0.04703
3033	0.06315	0.13220	0.06577	0.05034	0.06052
Mean	0.06866	0.07355	0.08700	0.05863	**0.05454**
SD	0.02081	0.03127	0.03189	0.01144	0.00930

Table 7: Elasticity ablation comparisons against Full CCR.

Comparison	d_z	Raw p	Holm p
Full CCR vs. Feature	0.6318	0.21875	1.00000
Full CCR vs. Direct	0.9560	0.09375	0.56250
Full CCR vs. NoConflict	0.2567	0.56250	1.00000

from 0.05863 to 0.05454. However, the difference between the two CCR variants is not statistically significant across the six seeds.

G.2 Thermal Ablation

Using the routing configuration transferred from elasticity without retuning, we evaluated the same routing variants on the 100-sample mixed thermal development set. For thermal conduction, Full

Table 8: Seed-wise thermal DEV-100 ablation results.

Seed	Base	Feature	Direct	NoConflict	Full CCR
123	0.02052	0.02866	0.01582	0.01743	0.01950
456	0.06273	0.01742	0.01697	0.01803	0.02548
789	0.01826	0.03502	0.02244	0.03225	0.02365
1011	0.03969	0.03157	0.03764	0.02030	0.01680
2022	0.02665	0.01891	0.01893	0.01833	0.02133
3033	0.02646	0.02644	0.04878	0.02085	0.02170
Mean	0.03238	0.02634	0.02676	**0.02120**	0.02141
SD	0.01663	0.00697	0.01341	0.00558	0.00305

CCR gives a mean error of 0.02141, compared with 0.02634 for Feature and 0.02676 for Direct. This corresponds to reductions of 18.70% and 20.00%, respectively. CCR-NoConflict gives a mean error of 0.02120, which is very close to Full CCR. The corresponding paired comparisons are reported in Table 9. Across the six thermal runs, Full CCR gives lower error for seeds 789 and 1011, while CCR-NoConflict gives lower error for the other four seeds. The paired comparison gives $d_z = -0.0382$ and $p = 0.9375$, showing no clear difference between the two variants for thermal conduction. Overall, the two bounded routing variants have lower mean errors than Feature and Direct for both PDEs. The conflict gate further reduces the mean error for elasticity, but does not show a clear improvement for thermal conduction.

Table 9: Thermal ablation comparisons against Full CCR.

Comparison	d_z	Raw p	Holm p
Full CCR vs. Feature	0.5658	0.21875	1.00000
Full CCR vs. Direct	0.3626	0.50000	1.00000
Full CCR vs. NoConflict	-0.0382	0.93750	1.00000