GEOSLICE-MGN: ANALYSIS-GUIDED GEOMETRIC ROUTING FOR IMPACT RESPONSE PREDICTION IN REALISTIC STRUCTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

Localized impact on complex structures produces high-gradient responses near the impact region and long-range structural coupling. Predicting these responses requires effective representations of loading position, structural boundaries, and local mesh geometry. We examine these geometric dependencies through finite-plate mechanics. Modal response analysis shows the roles of impact-relative and structure-relative positions. Analysis of bending and stretching strains shows that local spatial variations remain closely related to neighborhood directions and distances under finite deflection. Based on these relationships, we propose GeoSlice-MGN, which uses impact-relative geometry and bounding-box-relative position to modulate content-driven node-to-slice assignment. This makes load–structure geometry directly available to global routing in addition to its role in node features. The model retains local MeshGraphNet propagation before and after global interaction and augments slice states with relative soft occupancy. We also publicly release AutoPanelImpact, an open benchmark of 1,500 parameterized nonlinear impact cases on three real automotive floor panels, with paired displacement and stress fields under varying impact conditions. GeoSlice-MGN achieves the lowest average error on all six displacement–stress test metrics across the three panels. Experiments on BumperBeam and Plasticity further evaluate its applicability to coupled-field collision and deformation-sequence prediction.

1 INTRODUCTION

Localized impact on thin-walled automotive structures involves contact, large deformation, and material nonlinearity, typically requiring high-fidelity finite element (FE) analysis (Hallquist, 2006). Parametric studies become expensive when these simulations must be repeated for different impact locations, velocities, and impactor properties. Learned surrogates provide a computationally efficient alternative by directly mapping structural geometry and loading conditions to response fields. The state of peak deformation is especially relevant for automotive floor panels. Displacement at this event informs assessments of intrusion, residual clearance, and component interference, while the accompanying stress field characterizes local loading. We therefore study joint prediction of three-dimensional nodal displacements and shell-element von Mises effective stresses at this state from the initial shell mesh and impact conditions.

Vehicle-derived floor panels contain holes, stiffening features, and irregular boundaries. Moving the impact changes its spatial relationship to these features, even when the mesh connectivity remains fixed. A surrogate must therefore represent loading position together with structural geometry and capture local deformation, stress concentrations, and the far-field response induced by load transfer (Jones, 2012; Wierzbicki & Abramowicz, 1983).

Graph networks naturally model local interactions through FE connectivity (Pfaff et al., 2021; Maurizi et al., 2022; Würth et al., 2024), but long-range information must traverse multiple message-passing hops, limiting the effective receptive field of finite-depth models (Alon & Yahav, 2021; Topping et al., 2022). Latent slicing, as introduced in Transolver (Wu et al., 2024), enables global interaction by softly aggregating node features into compact slice states, updating them, and mapping the information back to the mesh. The assignments determine how node features are grouped and how

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

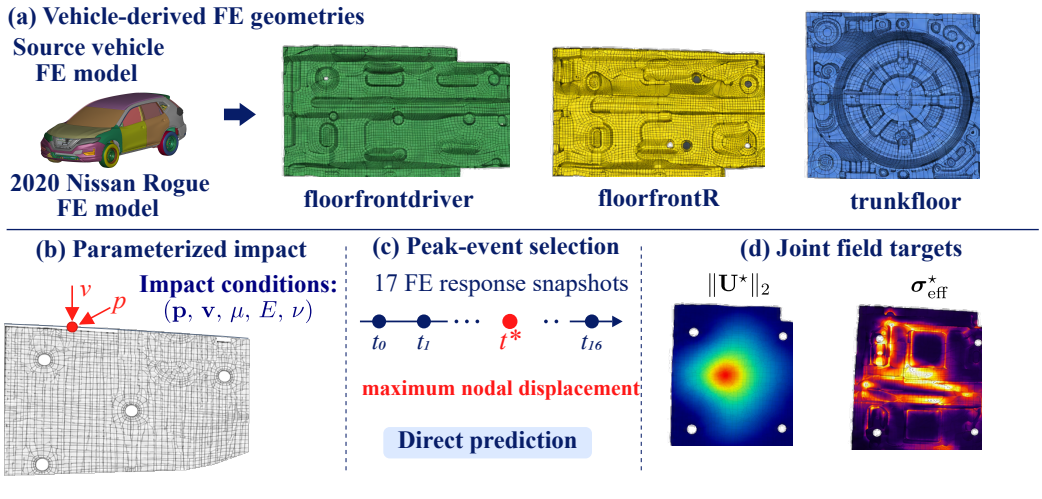


Figure 1: The `AutoPanelImpact` dataset and prediction task. (a) Three floor panels are extracted from a public full-vehicle FE model. (b) Impact conditions are varied. (c) The peak event $t^\star$, corresponding to the maximum displacement, is identified among 17 states. (d) Nodal displacements and shell-element effective stresses at this state are jointly predicted. Plots show FE ground truth.

global information returns to each node. We examine how latent grouping can reflect changing impact–structure relationships while retaining FE-local interactions.

We use a finite thin plate as an analytical reference to identify geometric information for slice assignment and local propagation. Its modal response depends jointly on observation position, load position, and boundaries, while the dependence of bending stress on spatial derivatives of deflection reveals the roles of neighborhood directions and spacings (Leissa, 1969; Bower, 2025). These relationships motivate the geometric representations used at the two interaction scales: impact-relative and structure-relative positions for latent global grouping, and edge vectors and lengths for FE-local propagation.

Based on this analysis, we introduce GeoSlice-MGN, a local–global–local network. Local MGN layers encode the mesh and loading conditions; a geometric branch then modulates content-driven node-to-slice logits through a gated residual, allowing the current load geometry to shape global grouping. After slice interaction, local refinement integrates the returned information before joint displacement–stress decoding. We also encode relative soft occupancy so that slice states retain information about total assignment weight alongside their average features.

To evaluate models systematically across varying impact conditions and realistic engineering geometries, we construct `AutoPanelImpact` from the public 2020 Nissan Rogue FE model (Center for Collision Safety and Analysis, 2024), comprising 1,500 nonlinear impact cases on three automotive floor panels with paired displacement and stress fields (Fig. 1). Loading variations within each panel and structural differences across panels support evaluation of conditional prediction, multi-panel modeling, and cross-panel adaptation. GeoSlice-MGN achieves the lowest mean test errors for both displacement and stress across all three automotive panels. Controlled variants, ablations, and routing diagnostics characterize the effects of its geometric routing and occupancy-aware design. Additional experiments on BumperBeam (AIRBORNEPANDA, 2026) and Plasticity (Li et al., 2023a) assess applicability to other collision and deformation tasks.

We make the following contributions:

- **Mechanics-guided geometric design.** We analyze finite-plate response and strain relationships to inform the geometric inputs used in global routing and local mesh propagation.
- **GeoSlice-MGN for joint-field prediction.** GeoSlice-MGN connects local encoding and refinement through geometrically modulated slice assignments and supplements slice states with occupancy information for peak displacement–stress prediction.

- **A benchmark on vehicle-derived floor-panel geometries.** `AutoPanelImpact` combines three floor panels extracted from a production-vehicle FE model with varying impact conditions and paired displacement–stress fields. The panels retain holes, stiffening features, and irregular boundaries, supporting evaluation on automotive structural geometries.

2 Related Work

Graph-based physical modeling. Graph neural networks represent discretized physical systems through particle neighborhoods or mesh connectivity (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021). This formulation supports structural-field prediction (Maurizi et al., 2022; Würth et al., 2024) and contact, impact, and crash modeling (Allen et al., 2023; Kim et al., 2025; Li et al., 2026; Elrefaie et al., 2026). Purely local message passing, however, requires successive hops to communicate between distant regions (Alon & Yahav, 2021; Topping et al., 2022). To extend this range, multiscale methods propagate information through mesh hierarchies (Fortunato et al., 2022; Cao et al., 2023; Yu et al., 2024; Würth et al., 2025), while local–global architectures combine neighborhood computation with global attention (Rampášek et al., 2022; Patrignani & Pinho, 2026; Iparraguirre et al., 2026).

Neural operators and latent global interaction. Neural operators learn mappings between input and response fields (Lu et al., 2021; Li et al., 2021). Extensions accommodate irregular geometries through coordinate transformations, graph integral transforms, and attention-based architectures with geometric gating (Li et al., 2023a;b; Hao et al., 2023). Related work compresses high-dimensional states for temporal attention or latent-space operator learning (Han et al., 2022; Wu et al., 2023). For spatial global interaction, Transolver aggregates physical points into latent states through soft slicing (Wu et al., 2024), with subsequent extensions improving scalability (Luo et al., 2025). GeoTransolver combines slice self-attention with cross-attention to shared geometry and global context built using multi-scale ball queries (Adams et al., 2026). It has also been applied to bumper-beam and vehicle crash dynamics (Akhare et al., 2026). Our work focuses on directly conditioning node-to-slice assignment logits with mechanics-guided load–structure geometry while retaining FE-local propagation.

3 GeoSlice-MGN

We jointly predict displacement and effective stress at peak deformation. For `AutoPanelImpact`, given an FE mesh $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with initial node coordinates and shell-element connectivity, and impact conditions $\mathbf{c} = (\mathbf{p}, \mathbf{v}, \mu, E, \nu)$, the prediction takes the form:

$$\mathcal{F}_\theta : (\mathcal{G}, \mathbf{c}) \longmapsto \big(\widehat{\mathbf{U}}^\star, \widehat{\boldsymbol{\sigma}}^\star_{\text{eff}}\big). \tag{1}$$

Here, $\mathbf{p}$ and $\mathbf{v}$ denote impact location and initial velocity. μ denotes the impactor mass ratio, implemented by scaling its density while keeping the impactor geometry fixed. E and ν are the impactor's Young's modulus and Poisson's ratio. Among the 17 saved states of each case, $t^\star$ is the time at which the maximum displacement magnitude over valid nodes is largest. Both target fields are taken from this event: $\mathbf{U}^\star \in \mathbb{R}^{N_v \times 3}$ contains nodal displacements, and $\boldsymbol{\sigma}^\star_{\text{eff}} \in \mathbb{R}^{N_e}$ contains shell-element stresses. Here, N_v and N_e are the numbers of nodes and shell elements. Appendix A provides further target definitions; the external tasks follow the settings in Appendices E.1 and E.2.

3.1 Overview

GeoSlice-MGN connects local MGN encoding and refinement through geometry-conditioned slicing and occupancy-aware global interaction (Fig. 2). Both local propagation and global grouping require geometric inputs. We first motivate these inputs through mechanical analysis, then describe how the network uses them to predict the two response fields.

3.2 Mechanics-Guided Geometric Representations

We identify geometric dependencies in the response of a finite plate, then examine whether they persist when mid-surface stretching is included. Consider a homogeneous, isotropic, simply supported

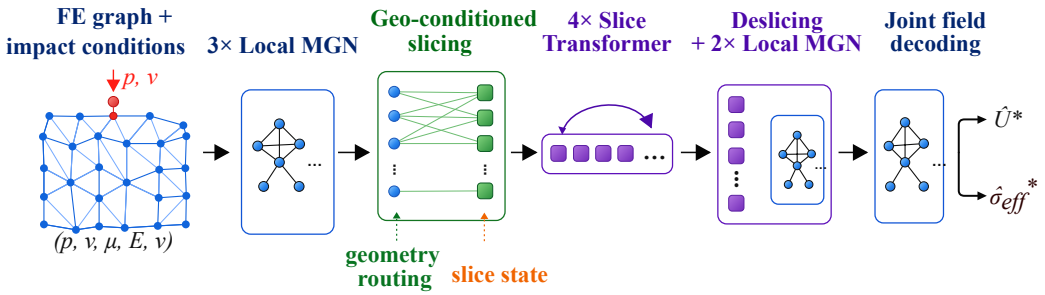


Figure 2: The local–global–local graph architecture of GeoSlice-MGN. Impact-relative geometry modulates node-to-slice routing through a gated residual, and the occupancy representation augments slices with assignment-weight information. Node aggregation and deslicing share the same soft assignment weights. After information is mapped back to nodes, local refinement precedes joint decoding of displacement and effective stress at $t^\star$.

rectangular plate with small deflections and linear elasticity. For an initially stationary, undamped system under a point load at $\mathbf{p}$, the transverse deflection is (Leissa, 1969; Donaldson, 2006)

$$w(\mathbf{x}, t) = \sum_{m,n\geq 1} \frac{\phi_{mn}(\mathbf{x})\phi_{mn}(\mathbf{p})}{M_{mn}\omega_{mn}} \int_0^t F(\tau)\sin[\omega_{mn}(t-\tau)]\ \mathrm{d}\tau. \tag{2}$$

Here, $\mathbf{x}$ is the observation location, $F(t)$ is the resultant transverse force, and M_{mn} and ω_{mn} are the modal mass and natural angular frequency. For side lengths L_x, L_y, the mode shape is $\phi_{mn}(x, y) = \sin(m\pi x/L_x)\sin(n\pi y/L_y)$. Equation (2) separates load history from spatial dependence: $\phi_{mn}(\mathbf{p})$ determines how strongly the load excites each mode, while $\phi_{mn}(\mathbf{x})$ determines its contribution at the observation point. Appendix B gives the derivation and finite-area load extension.

Let $\alpha_m = m\pi/L_x$. Then $\sin(\alpha_m x)\sin(\alpha_m p_x) = \frac{1}{2}[\cos(\alpha_m(x-p_x)) - \cos(\alpha_m(x+p_x))]$, with an analogous expression in the y direction. Both relative coordinates and positions within the structure enter the response: for a fixed finite domain and boundary conditions, identical node–load offsets can still produce different responses when the two points occupy different regions of the structure. This motivates representing both the node's position relative to the impact location and its position within the finite structure.

The strain relationships also motivate the local geometric inputs. In small-deflection bending, strains depend on second-order spatial variations of deflection, whose discrete representation is sensitive to neighborhood direction and spacing. When mid-surface stretching becomes relevant, the von Kármán approximation for small strains and moderate rotations retains bending strains and adds in-plane displacement gradients and quadratic transverse-gradient terms (Bower, 2025). These terms also depend on direction and spacing: a fixed displacement difference implies different gradients at different separations, and differently oriented edges provide different gradient projections. Thus, within this approximation, neighborhood directions and scales remain relevant when stretching is included. The coupled response still depends on load distribution and finite-domain boundaries.

This analysis motivates the ten-dimensional routing descriptor in Eq. (5): impact-relative vectors and distances describe node–load geometry, while bounding-box positions provide a coarse structural reference. Local MGN uses FE connectivity, edge vectors, and lengths to represent neighborhood directions and scales. These dependencies motivate the feature categories; we evaluate the specific descriptor empirically. Appendix B provides the detailed derivation.

3.3 Local MGN Encoding

We construct the local FE graph by adding bidirectional edges along each shell-element perimeter to encode local geometry. This defines the adjacency set $\mathcal{A}$. Each node is initialized from its spatial coordinates and task-specific conditioning features. Continuous inputs are normalized using statistics computed from the training set.

For a directed graph edge $(i, j) \in \mathcal{A}$, we define

$$\mathbf{r}_{ij} = \widetilde{\mathbf{x}}_j - \widetilde{\mathbf{x}}_i, \qquad \mathbf{f}_{ij} = [\mathbf{r}_{ij},\, \|\mathbf{r}_{ij}\|_2] \in \mathbb{R}^4, \tag{3}$$

where $\widetilde{\mathbf{x}}_i$ denotes normalized node coordinates. The vector and its length encode the neighborhood direction and scale motivated by the strain analysis above.

Node and edge encoders map these inputs to a C-dimensional latent space. Three MGN layers perform edge updates, neighborhood aggregation, and node updates to obtain node representations $\mathbf{h}_i$ (Battaglia et al., 2016; Pfaff et al., 2021). Appendix C details encoding and propagation.

3.4 Impact-Conditioned Node-to-Slice Routing

The locally encoded states $\mathbf{h}_i$ exchange long-range information through the soft-slicing mechanism of Transolver (Wu et al., 2024). We condition its assignments jointly on node content, impact-relative geometry, and structural position. For H_s routing heads with S slices each, separate projections of the normalized node state produce content logits and the node features to be aggregated:

$$L_{ihs}^{\mathrm{content}} = [W_A\,\mathrm{LN}(\mathbf{h}_i)]_{hs}\,, \qquad \mathbf{v}_{ih} = [W_V\,\mathrm{LN}(\mathbf{h}_i)]_h\,, \tag{4}$$

where $L_{ihs}^{\mathrm{content}}$ is the content logit for assigning node i to slice s in head h, and $\mathbf{v}_{ih}$ is the corresponding node feature. A geometric branch adds a residual to the content logits so that the spatial relationship between each node and the current load directly informs grouping.

Let $\mathbf{x}'_i$ and $\mathbf{p}'$ denote the normalized node and impact coordinates obtained using the same transformation as in Section 3.3. We combine the impact-relative vector, distances, and bounding-box positions into the ten-dimensional geometric input:

$$\mathbf{g}_i = \Bigg[\underbrace{\mathbf{x}'_i - \mathbf{p}',\, \|\mathbf{x}'_i - \mathbf{p}'\|_2,\, \|(\mathbf{x}'_i - \mathbf{p}')_{xy}\|_2}_{\text{impact-relative geometry}},\, \underbrace{\mathbf{x}'_{i,xy} - \mathbf{x}'_{\min,xy},\, \mathbf{x}'_{\max,xy} - \mathbf{x}'_{i,xy},\, q_i^x}_{\text{box-relative position}}\Bigg] \in \mathbb{R}^{10}, \tag{5}$$

The first five dimensions describe the direction and distance of a node relative to the impact location; the remaining five describe its position relative to the structure's XY bounding box. Here, q_i^x is the longitudinal position normalized by the bounding-box span. Coordinate preprocessing and feature definitions are given in Appendix C.

The geometric encoder f_θ generates a residual for each slice in each routing head, which we combine with the content logits through a learnable gate:

$$L_{ihs} = L_{ihs}^{\mathrm{content}} + \gamma_h f_\theta(\mathbf{g}_i)_{hs}, \qquad \gamma_h = \mathrm{sigmoid}(\alpha_h), \tag{6}$$

$$A_{ihs} = \mathrm{softmax}_s(L_{ihs}), \qquad \sum_{s=1}^{S} A_{ihs} = 1. \tag{7}$$

Here, γ_h controls the geometric residual strength in head h, and softmax is applied along the slice dimension. The resulting A_{ihs} determines each node feature's contribution to the slice states and is reused to map slice information back to nodes after global interaction. Initialization and tensor shapes are detailed in Appendix C.

3.5 Occupancy-Aware Global Slice Interaction

Given assignment weights A_{ihs}, we aggregate node features $\mathbf{v}_{ih}$ to construct slice states for global interaction. The soft occupancy and normalized state of a slice are

$$m_{hs} = \sum_{i=1}^{N_v} A_{ihs}, \qquad \mathbf{z}_{hs} = \frac{\sum_{i=1}^{N_v} A_{ihs}\mathbf{v}_{ih}}{m_{hs} + \epsilon}. \tag{8}$$

The normalized state $\mathbf{z}_{hs}$ summarizes average features, while m_{hs} records total assignment weight. We retain this additional information through an occupancy residual. Since $\sum_s A_{ihs} = 1$, uniform assignment gives a reference occupancy of N_v/S. We denote the clipped log-ratio of $m_{hs}/(N_v/S)$ by r_{hs} and encode it using a shared mapping ϕ:

$$\widetilde{\mathbf{z}}_{hs} = \mathbf{z}_{hs} + \phi(r_{hs}). \tag{9}$$

A slice Transformer applies attention among the S slices within each head to produce $\mathbf{Z}^G = \mathcal{T}_{\text{slice}}(\widetilde{\mathbf{Z}})$. It exchanges long-range information through this fixed number of latent states. Stop-gradient handling, numerical details, and the interaction implementation are described in Appendix C.

3.6 Deslicing, Local Refinement, and Joint-Field Decoding

After global interaction, we reuse A_{ihs} to map the updated slice states back to nodes and combine them with the local representations through a residual connection:

$$\mathbf{c}_{ih}^G = \sum_{s=1}^{S} A_{ihs}\mathbf{z}_{hs}^G, \qquad \mathbf{h}_i^{\text{ref},0} = \text{LN}\big[\mathbf{h}_i + W_D \,\text{Concat}_h\, \mathbf{c}_{ih}^G\big]\,. \tag{10}$$

Sharing A_{ihs} preserves the node–slice correspondence across aggregation and deslicing. Two further MGN layers refine the returned information along FE connectivity. The resulting node states are used to predict both fields: a node decoder predicts $\widehat{\mathbf{u}}_i^\star \in \mathbb{R}^3$, while an element decoder predicts $\widehat{\sigma}_{e,\text{eff}}^\star$ from node states and input features separately averaged over each element's vertices. Appendix C specifies the decoders, target normalization, and joint displacement–stress loss.

4 Experiments

4.1 The AutoPanelImpact Dataset

We construct `AutoPanelImpact` from three floor panels extracted from the public 2020 Nissan Rogue FE model (Center for Collision Safety and Analysis, 2024): `floorfrontdriver`, `floorfrontR`, and `trunkfloor`. The panels retain the source geometry, including holes, stiffening features, and irregular boundaries, together with the material constitutive models and shell-section definitions. These panels bring production-vehicle structural details into localized-impact prediction, extending the idealized block and plate settings represented in Table 5.

We generate 500 nonlinear LS-DYNA impact simulations per panel, giving $1{,}500$ cases in total. Within each panel, the mesh and support conditions remain fixed while impact location, speed, direction, and impactor parameters vary. The resulting cases provide controlled loading variations across three distinct structural layouts. Each case provides paired nodal displacement and shell-element effective stress fields, with both prediction targets taken from the same peak-deformation state. Complete cases are assigned to fixed training, validation, and test subsets of 400/50/50 cases per panel. Mesh, simulation, sampling, and data-processing details are provided in Appendix A.

4.2 Experimental Setup

For independent panel modeling, each model is trained three times on each panel, with checkpoints selected by validation performance. Test displacement and stress relative L_2 errors are reported as the mean and sample standard deviation across runs.

We compare against POD+MLP, DeepONet, MGN, Graph Neural Operator (GNO), GINO, Geo-FNO, and Transolver. We use additional controls to examine geometry injection and global interaction: MGN+Geo10 concatenates geometric inputs, Transolver+GeoRoute10 introduces geometric routing, and MGN+FullAttention performs attention over all nodes. Appendix D details these configurations.

4.3 Independent Training Results

We evaluate geometric routing on a second backbone by adding the ten-dimensional descriptor to Transolver's assignment logits through a gated residual. The resulting Transolver+GeoRoute10 lowers mean errors on five of the six metrics across the three panels, with the largest relative reductions on `trunkfloor`. Stress error increases on `floorfrontdriver`. These results support using load–structure geometry to guide slice assignments beyond the GeoSlice-MGN backbone.

Table 1: Test relative L_2 errors on the three panels (mean±sample standard deviation across three runs). Bold indicates the best result in each column; underlining indicates a lower mean for Transolver+GeoRoute10 than for Transolver. † denotes a control constructed in this work.

	`floorfrontdriver`		`floorfrontR`		`trunkfloor`	
Model	Displacement e_U	Stress e_σ	Displacement e_U	Stress e_σ	Displacement e_U	Stress e_σ
POD+MLP	0.2604 ± 0.0043	0.2597 ± 0.0007	0.2165 ± 0.0068	0.2581 ± 0.0049	0.5464 ± 0.0138	0.3966 ± 0.0023
DeepONet	0.2181 ± 0.0194	0.2434 ± 0.0067	0.1991 ± 0.0044	0.2452 ± 0.0082	0.4839 ± 0.0274	0.3971 ± 0.0048
GNO	0.1943 ± 0.0045	0.2158 ± 0.0054	0.1727 ± 0.0055	0.2153 ± 0.0013	0.4123 ± 0.0349	0.3997 ± 0.0058
GINO	0.1871 ± 0.0072	0.2090 ± 0.0011	0.1861 ± 0.0056	0.2234 ± 0.0013	0.3702 ± 0.0055	0.3489 ± 0.0068
Geo-FNO	0.1904 ± 0.0056	0.2393 ± 0.0045	0.1872 ± 0.0094	0.2521 ± 0.0108	0.4086 ± 0.0288	0.3899 ± 0.0106
MGN	0.1854 ± 0.0042	0.2229 ± 0.0063	0.1766 ± 0.0061	0.2300 ± 0.0025	0.4003 ± 0.0185	0.3940 ± 0.0079
MGN+Geo10†	0.1856 ± 0.0066	0.2219 ± 0.0015	0.1659 ± 0.0063	0.2293 ± 0.0017	0.3726 ± 0.0113	0.3846 ± 0.0053
Transolver	0.1710 ± 0.0018	0.1876 ± 0.0024	0.1722 ± 0.0094	0.2058 ± 0.0026	0.3656 ± 0.0295	0.3769 ± 0.0108
Transolver+GeoRoute10†	$\underline{0.1640} \pm 0.0065$	0.1951 ± 0.0039	$\underline{0.1594} \pm 0.0068$	$\underline{0.1990} \pm 0.0031$	$\underline{0.3130} \pm 0.0144$	$\underline{0.3439} \pm 0.0097$
MGN+FullAttention†	0.1681 ± 0.0125	0.1889 ± 0.0028	0.1740 ± 0.0048	0.1961 ± 0.0019	0.3682 ± 0.0142	0.3676 ± 0.0042
GeoSlice-MGN	$\mathbf{0.1506 \pm 0.0049}$	$\mathbf{0.1825 \pm 0.0042}$	$\mathbf{0.1521 \pm 0.0061}$	$\mathbf{0.1834 \pm 0.0041}$	$\mathbf{0.2952 \pm 0.0152}$	$\mathbf{0.3028 \pm 0.0143}$

Table 2: Test relative L_2 errors for combinations of core components (three seeds, mean±sample standard deviation). Standard deviations reflect variation due to training randomness.

Configuration	Displacement e_U	Stress e_σ
MGN	0.1854 ± 0.0042	0.2229 ± 0.0063
MGN + standard slicing	0.1699 ± 0.0101	0.1893 ± 0.0032
MGN + conditioned slicing	0.1732 ± 0.0227	0.2083 ± 0.0434
Standard slicing + occupancy	0.1648 ± 0.0059	0.1876 ± 0.0041
GeoSlice-MGN (full model)	0.1506 ± 0.0049	0.1825 ± 0.0042

The complete GeoSlice-MGN model achieves the lowest mean errors on all six metrics. On `trunkfloor`, its errors are a further 5.7% and 12.0% below those of Transolver+GeoRoute10, the best-performing control on that panel. Figure 3 shows a field-level comparison for one test case; Appendices G and H provide spatial statistics and full-field comparisons. Single-case inference takes approximately 15–26 ms under the timing protocol in Appendix J.

Appendix I further examines adaptation to a new panel. Fine-tuning a source-panel model with a small number of target cases improves target-panel predictions, supporting reuse of learned representations when target data are limited.

4.4 Design Ablations and Routing Diagnostics

To assess the combined contributions of global slicing, geometric conditioning, and occupancy, Table 2 compares five configurations on `floorfrontdriver`. Supplementary ablations and configuration details are provided in Appendix F.

Table 2 shows that global slicing lowers mean prediction errors relative to local MGN. When both configurations retain the auxiliary occupancy term, geometrically conditioned slicing improves on

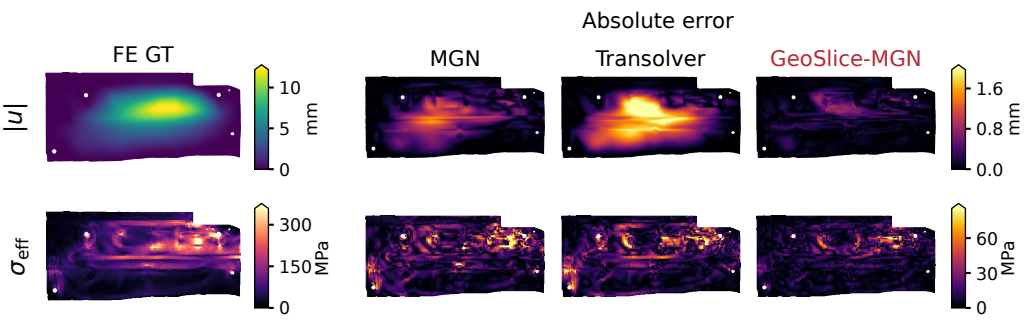


Figure 3: Ground truth and absolute errors on `floorfrontdriver`. Top: displacement magnitude. Bottom: effective stress. The first column shows ground truth; the remaining three show errors.

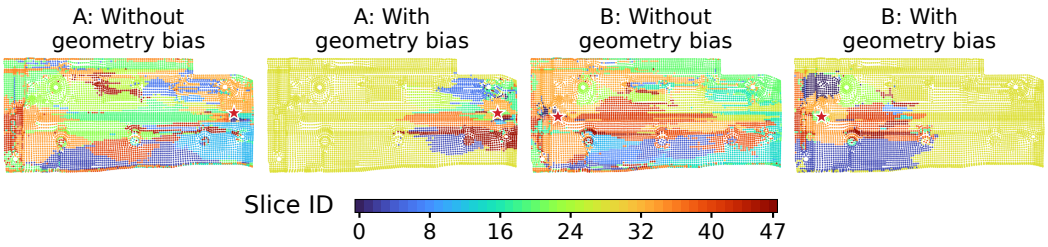


Figure 4: A and B are two impact cases on the same panel. For each case, slice assignments are compared with and without the explicit geometric bias. Colors indicate the slice with the highest assignment probability within the same attention head, and red stars mark impact locations. Each pair uses identical model parameters and pre-routing node states.

standard slicing for displacement and stress, with a larger relative gain in displacement. Without this term, conditioned slicing has higher mean errors and greater variability across seeds than standard slicing on this panel. The full model gives the lowest mean errors for both fields, supporting the combination of local propagation and geometrically conditioned global interaction.

Spatial grouping through geometric conditioning. Figure 4 shows how grouping changes when the explicit geometric bias is removed while model parameters and pre-routing node states are held fixed within each case. Without the bias, assignments to different slices are more interspersed. With it, the region farthest from impact is predominantly assigned to one slice, while the impact neighborhood retains several assignments. The region assigned to multiple slices follows the load from the right side in A to the left side in B. This comparison complements the error-based ablations by showing how the geometric branch reorganizes assignments according to node–impact relationships.

4.5 Shared Modeling Across Multiple Panels

We assess whether a single model can predict displacement and stress across three vehicle-derived panels with distinct geometries and mesh connectivity. Each shared model uses one set of learned parameters and is trained on the combined cases from all three panels, with normalization statistics computed exclusively from this joint training set. Table 3 reports panel-wise test errors across three runs; Appendix C.3 gives the training and evaluation procedure.

Table 3: Panel-wise test relative L_2 errors for shared models (mean$\pm$sample standard deviation across three independent runs). Bold indicates the lower mean within each panel.

	Displacement e_U		Stress e_σ	
Panel	Transolver	GeoSlice-MGN	Transolver	GeoSlice-MGN
floorfrontdriver	0.1516 ± 0.0064	$\mathbf{0.1473 \pm 0.0017}$	0.2027 ± 0.0041	$\mathbf{0.1835 \pm 0.0023}$
floorfrontR	0.1405 ± 0.0033	$\mathbf{0.1343 \pm 0.0056}$	0.2129 ± 0.0038	$\mathbf{0.1888 \pm 0.0038}$
trunkfloor	0.3051 ± 0.0112	$\mathbf{0.2870 \pm 0.0109}$	0.3268 ± 0.0072	$\mathbf{0.2941 \pm 0.0086}$

Using the same learned parameters across all three panels, GeoSlice-MGN achieves lower mean displacement and stress errors than the shared Transolver on every panel. These results support joint modeling of distinct automotive panel geometries with a single surrogate. Evaluation uses held-out loading cases for the three geometries included in training.

4.6 Evaluation on Public Datasets

We evaluate the architecture on two public tasks with adapted inputs and prediction targets. On BumperBeam, the ten-dimensional geometric descriptor uses the cylinder-axis reference as the loading location, and the model predicts peak-state displacement and stress. On Plasticity (Li et al., 2023a), a seven-dimensional descriptor combines logical coordinates, the die profile, and domain-edge indicators; the decoder outputs 20 coordinate and displacement frames in one pass.

Both adaptations retain local encoding and refinement, gated geometric modulation of node-to-slice assignment logits, and occupancy-augmented slice interaction. Task-specific inputs, descriptors, decoders, and model configurations are detailed in Appendices E.1 and E.2. Within each dataset, all models are trained from scratch on the same case splits. Table 4 reports test errors.

Table 4: Test relative L_2 errors on public datasets, reported as mean±sample standard deviation across three independent runs. The joint metric for Plasticity includes deformed coordinates and displacements. Bold indicates the lowest mean in each column.

	BumperBeam (test)		Plasticity (test)	
Model	Displacement ↓	Stress ↓	Joint ↓	Displacement ↓
MGN	0.04731 ± 0.00147	0.27531 ± 0.00491	0.035064 ± 0.000448	0.744861 ± 0.008081
GNO	0.04861 ± 0.00365	0.26969 ± 0.00672	0.027972 ± 0.000632	0.599464 ± 0.011095
Geo-FNO	0.06617 ± 0.00595	0.35003 ± 0.01332	0.008725 ± 0.000229	0.206944 ± 0.005673
GINO	0.05889 ± 0.00572	0.33091 ± 0.02429	0.007742 ± 0.000111	0.178231 ± 0.002653
Transolver	$\mathbf{0.04092 \pm 0.00262}$	0.30632 ± 0.01447	0.008282 ± 0.000957	0.189090 ± 0.021305
GeoSlice-MGN	0.04154 ± 0.00156	$\mathbf{0.24522 \pm 0.00764}$	$\mathbf{0.006014 \pm 0.000757}$	$\mathbf{0.132700 \pm 0.017770}$

On BumperBeam, GeoSlice-MGN has a displacement error close to Transolver's and the lowest mean stress error among the compared models, reducing stress error by 19.95% and 9.07% relative to Transolver and GNO, respectively. On Plasticity, GeoSlice-MGN achieves the lowest mean joint and displacement errors among the compared models under the common full-sequence training protocol, improving on GINO by 22.32% and 25.55%, respectively. These results support using the geometric routing and slice-interaction architecture for both coupled-field collision prediction and deformation-sequence prediction with task-specific geometric descriptors.

5 Conclusion

We presented GeoSlice-MGN for joint prediction of peak displacement and effective stress under localized impact. Guided by finite-plate analysis, the model conditions latent node-to-slice routing on impact- and structure-relative geometry. Local MeshGraphNet propagation represents neighborhood interactions. We also publicly release AutoPanelImpact, which contains 1,500 nonlinear impact simulations on three floor panels extracted from a full-vehicle FE model. The panels retain holes, stiffening features, and irregular boundaries. GeoSlice-MGN achieves the lowest mean errors on all six main test metrics for this dataset. Shared-model experiments show that a single model can predict responses across all three panel geometries, and task-adapted experiments on BumperBeam and Plasticity further support the architecture's applicability to collision and deformation-sequence prediction. Ablations support the contribution of geometric routing within the full architecture. Routing diagnostics show how it reorganizes latent grouping according to node–impact relationships. The main evaluation focuses on held-out loading cases for fixed panel geometries. Future work will extend the cross-panel study to a broader range of unseen structures, support conditions, and loading regimes. We will also investigate richer representations of structural boundaries and contact geometry, as well as extending peak-state prediction to the full transient impact response.

AI Use Statement

We used generative AI tools to help draft and revise parts of the manuscript, edit the language, organize the text, and check LaTeX formatting. We also used these tools to help write auxiliary scripts for cross-checking experimental records. The authors reviewed the AI-assisted content against source files, code, and saved experimental records. We take responsibility for the final content of this work, including all text, code, and other artifacts produced with the assistance of generative AI.

Ethics Statement

This work uses public automotive FE models and computational simulations. It does not involve human subjects or personal data.

Reproducibility Statement

Code for independent training and evaluation of GeoSlice-MGN on the three panels is publicly available at `https://anonymous.4open.science/r/GeoSliceMGN/`, including the model implementation, data preprocessing, fixed splits, training configurations, and evaluation scripts. The `AutoPanelImpact` dataset is publicly available at `https://anonymous.4open.science/r/underbody-impact-data-7CA4`. Section 4.1 specifies the dataset size and fixed splits, and Appendix A describes simulation, sampling, and result-export settings. Appendix C defines target normalization and loss functions, and Section 4 describes the evaluation metrics and test protocol. The appendices also document model configurations, baseline adaptations, design ablations, spatial error analyses, cross-panel adaptation, and computational efficiency. The pinned version, generation project, and training and evaluation settings for the public BumperBeam data are given in Appendix E.1.

References

Corey Adams, Rishikesh Ranade, Ram Cherukuri, and Sanjay Choudhry. GeoTransolver: Learning physics on irregular domains using multi-scale geometry aware physics attention transformer, 2026. URL https://arxiv.org/abs/2512.20399.

AIRBORNEPANDA. BumperBeamCrashExample. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/AIRBORNEPANDA/BumperBeamCrashExample/tree/78f1302fba4d75926c6fdd22d01d5133efc14742.

Deepak Akhare, Mohammad Amin Nabian, Corey Adams, Sudeep Chavare, and Sanjay Choudhry. High-fidelity industrial crash dynamics prediction via geometry-aware operator learning with memory-efficient low-rank attention, 2026. URL https://arxiv.org/abs/2605.27758.

Kelsey R. Allen, Yulia Rubanova, Tatiana Lopez-Guevara, William Whitney, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Tobias Pfaff. Learning rigid dynamics with face interaction graph networks. In *International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/10983.

Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=i80OPhOCVH2.

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4502–4510, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/3147da8ab4a0437c15ef51a5cc7f2dc4-Abstract.html.

Allan F. Bower. Applied mechanics of solids: Simplified versions of general shell theory—flat plates and membranes. Online textbook, Section 10.6, 2025. URL https://solidmechanics.org/Text/Chapter10_6/Chapter10_6.php. Accessed September 18, 2026.

Yadi Cao, Menglei Chai, Minchen Li, and Chenfanfu Jiang. Efficient learning of mesh-based physical simulation with bi-stride multi-scale graph neural network. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 3541–3558. PMLR, 2023. URL https://proceedings.mlr.press/v202/cao23a.html.

Center for Collision Safety and Analysis. 2020 Nissan Rogue sport utility vehicle model, version 3. George Mason University, 2024. URL https://www.ccsa.gmu.edu/models/2020-nissan-rogue/. Version 3 released August 2024.

Bruce K. Donaldson. *Introduction to Structural Dynamics*. Cambridge University Press, 2006. doi: 10.1017/cbo9780511618086.

Mohamed Elrefaie, Dule Shu, Matthew Klenk, and Faez Ahmed. CarCrashNet: A large-scale dataset and hierarchical neural solver for data-driven structural crash simulation. *arXiv preprint arXiv:2605.07098*, 2026. URL https://arxiv.org/abs/2605.07098.

Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter Battaglia. Multiscale MeshGraphNets. In *ICML 2022 2nd AI for Science Workshop*, 2022. URL https://openreview.net/forum?id=G3TRIsmMhhf.

John O. Hallquist. *LS-DYNA Theory Manual*. Livermore Software Technology Corporation, 2006. URL https://www.dynalook.com/conferences/international-conf-2006/theory-manual-2006.pdf.

Xu Han, Han Gao, Tobias Pfaff, Jian-Xun Wang, and Li-Ping Liu. Predicting physics in mesh-reduced space with temporal attention. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=XctLdNfCmP.

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. URL https://proceedings.mlr.press/v202/hao23c.html.

J.S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018. doi: 10.1016/j.jcp.2018.02.037.

Mikel M. Iparraguirre, Iciar Alfaro, David Gonzalez, and Elias Cueto. MeshGraphNet-Transformer: Scalable mesh-based learned simulation for solid mechanics. *arXiv preprint arXiv:2601.23177*, 2026. URL https://arxiv.org/abs/2601.23177.

Norman Jones. *Structural Impact*. Cambridge University Press, 2nd edition, 2012. doi: 10.1017/CBO9780511820625.

Jiyong Kim, Jangseop Park, Nayong Kim, Younyeol Yu, Kiseok Chang, Chang-Seung Woo, Sunwoong Yang, and Namwoo Kang. Physics-constrained graph neural networks for spatio-temporal prediction of drop impact on OLED display panels. *Expert Systems with Applications*, 274:126907, 2025. doi: 10.1016/j.eswa.2025.126907.

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. URL https://www.jmlr.org/papers/v24/21-1524.html.

A. W. Leissa. Vibration of plates. Technical Report NASA-SP-160, National Aeronautics and Space Administration, 1969. URL https://ntrs.nasa.gov/citations/19700009156.

Haoran Li, Yingxue Zhao, Haosu Zhou, Tobias Pfaff, and Nan Li. A graph neural network surrogate model for mesh-based crashworthiness prediction of vehicle panel components. *Results in Engineering*, 30:110925, 2026. doi: 10.1016/j.rineng.2026.110925.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020. URL https://arxiv.org/abs/2003.03485.

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://openreview.net/forum?id=c8P9NQVtmnO.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023a. URL https://www.jmlr.org/papers/v24/23-0064.html.

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, 2023b. doi: 10.52202/075280-1556.

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*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5.

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. URL https://proceedings.mlr.press/v267/luo25o.html.

Marco Maurizi, Chao Gao, and Filippo Berto. Predicting stress, strain and deformation fields in materials and structures with graph neural networks. *Scientific Reports*, 12:21834, 2022. doi: 10.1038/s41598-022-26424-3.

Luca Patrignani and Silvestre T. Pinho. Graph neural networks with hybrid local-global attention for effective prediction of mechanical response in structures. *Computer Methods in Applied Mechanics and Engineering*, 452:118753, 2026. doi: 10.1016/j.cma.2026.118753.

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

Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14501–14515, 2022. URL https://neurips.cc/virtual/2022/poster/54958.

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. URL https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html.

Jake Topping, Francesco Di Giovanni, Benjamin P. Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7UmjRGzp-A.

Tomasz Wierzbicki and Wlodzimierz Abramowicz. On the crushing mechanics of thin-walled structures. *Journal of Applied Mechanics*, 50(4a):727–734, 1983. doi: 10.1115/1.3167137.

Haixu Wu, Tengge Hu, Huakun Luo, Jianmin Wang, and Mingsheng Long. Solving high-dimensional PDEs with latent spectral models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 37417–37438. PMLR, 2023. URL https://proceedings.mlr.press/v202/wu23f.html.

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. URL https://proceedings.mlr.press/v235/wu24r.html.

Tobias Würth, Niklas Freymuth, Clemens Zimmerling, Gerhard Neumann, and Luise Kärger. Physics-informed MeshGraphNets (PI-MGNs): Neural finite element solvers for non-stationary and nonlinear simulations on arbitrary meshes. *Computer Methods in Applied Mechanics and Engineering*, 429:117102, 2024. doi: 10.1016/j.cma.2024.117102.

Tobias Würth, Niklas Freymuth, Gerhard Neumann, and Luise Kärger. Diffusion-based hierarchical graph neural networks for simulating nonlinear solid mechanics. In *Advances in Neural Information Processing Systems*, volume 38, pp. 144338–144367, 2025. doi: 10.52202/085713-4336.

Youn-Yeol Yu, Jeongwhan Choi, Woojin Cho, Kookjin Lee, Nayong Kim, Kiseok Chang, Chang-Seung Woo, Ilho Kim, Seok-Woo Lee, Joon-Young Yang, Sooyoung Yoon, and Noseong Park. Learning flexible body collision dynamics with hierarchical contact mesh transformer. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=90yw2uM6J5.

A Construction and Characteristics of AutoPanelImpact

A.1 Finite Element Models and Contact Settings

We simulate impact in LS-DYNA using substructures extracted from the 2020 Nissan Rogue Version 3 model (Center for Collision Safety and Analysis, 2024), retaining their material constitutive models and shell-section definitions. The meshes contain 7,408 nodes and 7,374 shell elements for `floorfrontdriver`, 12,011 nodes and 12,055 elements for `floorfrontR`, and 14,440 nodes and 14,589 elements for `trunkfloor`.

The impactor is a rigid spherical shell modeled with `MAT_RIGID`, with radius 12.5 mm and thickness 0.1 mm. For impact location $\mathbf{p}$ and unit velocity direction $\widehat{\mathbf{v}}$, its initial center is $\mathbf{p} - (17.5\ \mathrm{mm})\widehat{\mathbf{v}}$, giving a nominal surface gap of 5 mm along the incident direction. Its initial angular velocity is zero. Automatic surface-to-surface contact couples the impactor and panel, with both static and dynamic friction coefficients set to 0.15. A body acceleration of 9810 mm/s^2 acts along global $+Z$. Simulations use the consistent tonne–mm–s–N unit system; exported displacements and stresses are in mm and MPa, respectively.

A.2 Case Sampling and Result Export

We sample each panel independently using a seven-dimensional constrained Latin hypercube design. The variables specify planar impact position, speed, mass scaling, two incidence angles, and impactor material category. Eligible impact locations are shell-element centers at least 80 mm from every topological boundary node in the XY plane. Normalized planar samples are mapped to the nearest unused eligible center, whose coordinate determines the impact height Z. We select the design using a normalized minimum interpoint-distance criterion.

Speed ranges from 1732.05 to 5196.15 mm/s. The factor $\mu \in [0.75, 1.25]$ scales the reference impactor density of 5.205×10^{-5} tonne/mm^3. The polar incidence angle lies in $[0^\circ, 15^\circ]$ relative to global $+Z$, and the azimuth lies in $[0^\circ, 360^\circ]$, measured from $+X$ toward $+Y$. We sample the impactor parameter pairs $(E, \nu) = (70000\ \mathrm{MPa}, 0.33)$, $(110000\ \mathrm{MPa}, 0.34)$, and $(210000\ \mathrm{MPa}, 0.30)$ in approximately balanced numbers. These parameters describe the rigid impactor; the panel constitutive models remain fixed.

LS-PrePost exports three-component nodal displacement and shell-element effective stress. We retain every tenth original state and the final state, giving 17 paired frames per case. The first 16 frames have a nominal interval of 2 ms; the last uses its actual saved timestamp. Element stress is the maximum von Mises stress over the through-thickness integration points. Node-validity and fill-value masks are retained with the fields. Peak selection uses valid nodes and identifies the discrete maximum among the retained states.

A.3 Dataset Characteristics and Comparison with Public Benchmarks

Table 5 compares public datasets by structure, geometry type, case count, and response fields. Plasticity, Deforming Plate, and Impact Plate use idealized block or plate geometries, while BumperBeam provides an automotive component benchmark. `AutoPanelImpact` extends localized-impact evaluation to three floor panels extracted from the public 2020 Nissan Rogue FE model (Center for Collision Safety and Analysis, 2024). These vehicle-derived geometries bring automotive floor-panel details into the prediction task by retaining holes, stiffening features, and irregular boundaries. CarCrashNet (Elrefaie et al., 2026) also covers automotive component and full-vehicle crash simulations. Its official repository lists dataset release as pending completion of peer review, so it is not included in Table 5.

Table 5: Comparison of publicly available structural deformation and impact datasets.

Dataset	Structure	Geometry type	Cases	Fields
Plasticity (Li et al., 2023a)	2D block	Idealized	987	x, u
Deforming Plate (Pfaff et al., 2021)	3D plate	Idealized	1,200	x, σ
Impact Plate (Yu et al., 2024)	2D plate	Idealized	2,400	x, σ
BumperBeam (AIRBORNEPANDA, 2026)	3D bumper beam	Engineering benchmark	131	u, σ
AutoPanelImpact (this work)	3D floor panels	Real vehicle geometry	1,500	u, σ

The table includes datasets with publicly accessible simulation data at the time of submission. Geometry type describes the structural geometry used in simulation. Real vehicle geometry denotes components extracted from a production-vehicle FE model. x, u, and σ denote position, displacement, and stress fields, respectively.

Within each panel, impact location, velocity, and impactor properties vary while the mesh and supports remain fixed. Moving the impact changes its spatial relationship to holes, stiffening features, and boundaries. This allows controlled evaluation of how a surrogate represents loading conditions together with structural geometry. Differences between panels support shared modeling and cross-panel adaptation. The 17 paired states per case provide displacement and stress at a common peak-deformation state for joint assessment of overall deformation and localized loading.

B Mechanical Rationale for the Geometric Design

We examine the mechanical basis for the geometric inputs. A simply supported plate under small deflections provides a reference for how load-relative and structural positions enter the response. We then relate local strains to neighborhood directions and spacings and examine whether these dependencies persist with mid-surface stretching. Finally, we connect the resulting dependencies to geometric routing and local message passing.

B.1 Finite-Plate Reference Problem and Finite-Area Contact Loading

Consider an initially flat rectangular plate $\Omega = [0, L_x] \times [0, L_y]$ with uniform thickness and homogeneous, isotropic material. These assumptions isolate the effects of load position and finite boundaries from those of initial curvature, thickness variation, and material heterogeneity. Let $\mathbf{x} = (x, y)$ be the observation position and $w(\mathbf{x}, t)$ the transverse mid-surface deflection. Plate thickness, density, Young's modulus, and Poisson's ratio are denoted by $h, \rho, E_{\mathrm{p}}, \nu_{\mathrm{p}}$.

Under Kirchhoff–Love kinematics, fibers initially normal to the mid-surface remain straight and normal after deformation. Neglecting transverse shear deformation and thickness stretching allows bending strains to be expressed through mid-surface motion. We begin with small deflections, no initial membrane forces, and linear elasticity, so deflection-induced mid-surface stretching is neglected. These kinematic and constitutive assumptions yield a linear transverse-response equation (Bower, 2025, Section 10.6.1). Neglecting damping and rotary inertia gives

$$\rho h w_{tt} + D\nabla^4 w = q_c(\mathbf{x}, t), \qquad D = \frac{E_{\mathrm{p}} h^3}{12(1-\nu_{\mathrm{p}}^2)}, \tag{11}$$

where D is the bending stiffness, ∇^4 is the biharmonic operator on the plate plane, and q_c is the transverse distributed load over a finite contact region $A_c(t) \subset \Omega$. To distinguish contact location from contact scale, we fix the contact center $\mathbf{p} = (p_x, p_y)$ inside the plate and write

$$q_c(\mathbf{x}, t) = F(t)\kappa(\mathbf{x} - \mathbf{p}; a(t)), \qquad \int_{A_c(t)} \kappa(\mathbf{x} - \mathbf{p}; a(t)) \, \mathrm{d}A = 1. \tag{12}$$

Here, $F(t)$ is the resultant transverse force, $a(t)$ describes contact-region size, and κ is a normalized load distribution supported on $A_c(t)$. Fixing the center while allowing the contact region to vary in size over time separates the effects of location and size. We prescribe both the force history and spatial distribution to examine structural response; feedback between contact pressure and deformation is outside this reference problem.

All four edges are simply supported: transverse displacement and normal bending moment vanish, while rotation is permitted. These conditions give finite supports and explicit mode shapes with which to distinguish load-relative and structural-position effects. They serve as an analytical reference and do not describe the mounting constraints of the automotive panels. The mode shapes are (Leissa, 1969, Section 4.1)

$$\phi_{mn}(x,y) = \sin\frac{m\pi x}{L_x}\sin\frac{n\pi y}{L_y}, \qquad m,n \geq 1. \tag{13}$$

Each mode shape ϕ_{mn} describes a spatial deformation pattern with time-dependent amplitude $q_{mn}(t)$. Substituting the modal expansion $w(\mathbf{x},t) = \sum_{m,n\geq 1} q_{mn}(t)\phi_{mn}(\mathbf{x})$ into Eq. (11) and using linearity and modal orthogonality yields independent vibration equations:

$$M_{mn}\ddot{q}_{mn} + M_{mn}\omega_{mn}^2 q_{mn} = Q_{mn}(t), \qquad Q_{mn}(t) = \int_{A_c(t)} q_c(\mathbf{x},t)\phi_{mn}(\mathbf{x})\,\mathrm{d}A, \tag{14}$$

where

$$\omega_{mn} = \pi^2\sqrt{\frac{D}{\rho h}}\left(\frac{m^2}{L_x^2}+\frac{n^2}{L_y^2}\right), \qquad M_{mn} = \frac{\rho h L_x L_y}{4}. \tag{15}$$

Substituting Eq. (12) into the modal projection gives

$$Q_{mn}(t) = F(t)\int_{A_c(t)} \kappa(\mathbf{x}-\mathbf{p};a(t))\,\phi_{mn}(\mathbf{x})\,\mathrm{d}A. \tag{16}$$

Equation (16) shows that excitation depends on the mode shape averaged over the contact region with weights κ. As that region contracts to $\mathbf{p}$ and $\kappa(\mathbf{x}-\mathbf{p};a)$ converges distributionally to $\delta^{(2)}(\mathbf{x}-\mathbf{p})$ as $a \to 0$, this average approaches the mode-shape value at the load point:

$$Q_{mn}(t) \longrightarrow F(t)\phi_{mn}(\mathbf{p}). \tag{17}$$

The point-load limit makes the spatial factors explicit. For a finite contact region, this approximation is justified only for modes that vary sufficiently little across the region. A small contact area alone is insufficient to justify it for all higher-order modes. We therefore use the limit to examine positional dependence, while Eq. (16) retains the finite-area description.

B.2 Relative and Structure-Relative Positions in the Modal Response

For an initially stationary plate, $w(\mathbf{x},0) = w_t(\mathbf{x},0) = 0$, the response contains only load-induced excitation. Together with the undamped assumption, this removes free initial motion and dissipative decay from the temporal kernel, making the spatial dependence explicit. In the point-load limit, Duhamel's integral (Donaldson, 2006, Section 7.6) gives

$$w(\mathbf{x},t) = \sum_{m,n\geq 1}\frac{\phi_{mn}(\mathbf{x})\phi_{mn}(\mathbf{p})}{M_{mn}\omega_{mn}}\int_0^t F(\tau)\sin[\omega_{mn}(t-\tau)]\,\mathrm{d}\tau. \tag{18}$$

Each modal contribution is thus a temporal integral multiplied by two spatial factors: $\phi_{mn}(\mathbf{p})$ gives the modal excitation weight at the load location, and $\phi_{mn}(\mathbf{x})$ gives the mode-shape value at the observation location.

With $\alpha_m = m\pi/L_x$, the modal factor in the x direction satisfies

$$\sin(\alpha_m x)\sin(\alpha_m p_x) = \frac{1}{2}\left[\cos(\alpha_m(x-p_x)) - \cos(\alpha_m(x+p_x))\right]. \tag{19}$$

The first term depends on $x-p_x$, which retains the signed position of the observation point relative to the load. The second depends on $x+p_x$ and therefore on where the pair lies within the fixed domain.

Two pairs can satisfy $x_1 - p_{x,1} = x_2 - p_{x,2}$ yet have different coordinate sums and modal weights. The same decomposition applies in y. Thus, even with fixed load history and physical conditions, the full relative coordinate generally does not determine the response:

$$w(\mathbf{x}, t) \neq \widetilde{\mathcal{F}}(\mathbf{x} - \mathbf{p}, t) \quad \text{(in general)}. \tag{20}$$

The finite domain and its boundary conditions generally break joint translational symmetry of the load and observation points. Identical node–impact offsets can therefore produce different responses in different structural regions. For finite-area contact, Eq. (16) replaces the load-point mode-shape value with a weighted regional average, while preserving dependence on position within the structure.

Accordingly, for node i, define

$$\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{p}. \tag{21}$$

The vector records direction and offset relative to impact, and its length records distance. The modal result motivates the first design consideration: the joint role of load-relative and structural positions. A structural reference distinguishes identical impact-relative offsets in different regions.

B.3 Local Strain, Stress, and Mesh Geometry

We next relate deflection to local strain. In the small-deflection pure-bending reference, the mid-surface has no membrane extension, while fibers above and below it stretch or compress. Strain therefore depends on curvature and the signed thickness coordinate ζ. Kirchhoff–Love kinematics give (Bower, 2025, Section 10.6.1)

$$\varepsilon_x = -\zeta w_{xx}, \qquad \varepsilon_y = -\zeta w_{yy}, \qquad \gamma_{xy} = -2\zeta w_{xy}. \tag{22}$$

Using linear elasticity and the plane-stress approximation, which neglects through-thickness stress, the corresponding bending stresses are

$$\begin{aligned} \sigma_x &= -\frac{E_\mathrm{p}\zeta}{1-\nu_\mathrm{p}^2}(w_{xx} + \nu_\mathrm{p} w_{yy}), \qquad \sigma_y = -\frac{E_\mathrm{p}\zeta}{1-\nu_\mathrm{p}^2}(w_{yy} + \nu_\mathrm{p} w_{xx}), \\ \tau_{xy} &= -\frac{E_\mathrm{p}\zeta}{1+\nu_\mathrm{p}} w_{xy}. \end{aligned} \tag{23}$$

Writing the deflection Hessian explicitly gives the response chain

$$\mathbf{H}_w = \begin{bmatrix} w_{xx} & w_{xy} \\ w_{xy} & w_{yy} \end{bmatrix}, \qquad F \longrightarrow w \longrightarrow \mathbf{H}_w \longrightarrow \text{bending stress}. \tag{24}$$

This chain connects bending stress to spatial curvature. A local predictor must interpret displacement or hidden-feature variations together with neighborhood geometry.

A second-derivative approximation illustrates the role of spacing. On a one-dimensional uniform grid with spacing ℓ, we have

$$w_{xx}(x_i) \approx \frac{w(x_i + \ell) - 2w(x_i) + w(x_i - \ell)}{\ell^2}. \tag{25}$$

For the same second difference in nodal displacement, changing ℓ changes the estimated curvature; halving ℓ increases it by a factor of four. Local field differences must therefore be interpreted using node spacing.

To illustrate the role of direction, consider an unstructured neighborhood around node i on a flat plate, with in-plane edge vector $\mathbf{d}_{ij} = (d_x, d_y)^\mathsf{T} = \mathbf{x}_j - \mathbf{x}_i$. For a locally smooth deflection field and a sufficiently small neighborhood, a second-order Taylor expansion gives

$$w_j - w_i = \nabla w_i^\mathsf{T} \mathbf{d}_{ij} + \frac{1}{2}\mathbf{d}_{ij}^\mathsf{T} \mathbf{H}_{w,i} \mathbf{d}_{ij} + O(\|\mathbf{d}_{ij}\|^3), \tag{26}$$

where $\mathbf{H}_{w,i} = \mathbf{H}_w(\mathbf{x}_i, t)$ and

$$\mathbf{d}_{ij}^\mathsf{T} \mathbf{H}_{w,i} \mathbf{d}_{ij} = d_x^2 w_{xx} + 2 d_x d_y w_{xy} + d_y^2 w_{yy} = \ell_{ij}^2 \mathbf{n}_{ij}^\mathsf{T} \mathbf{H}_{w,i} \mathbf{n}_{ij}. \tag{27}$$

Here, $\ell_{ij} = \|\mathbf{d}_{ij}\|_2$ and $\mathbf{n}_{ij} = \mathbf{d}_{ij}/\ell_{ij}$, with derivatives evaluated at node i. The factor ℓ_{ij}^2 sets the scale of the second-order variation, while $\mathbf{n}_{ij}$ selects a directional combination of w_{xx}, w_{yy}, w_{xy}. Edge vectors and lengths thus connect response differences to local curvature. Adjacency identifies neighbors but does not by itself supply their directions or separations. The second design consideration is therefore the need for neighborhood geometry in local strain representations.

The local bending reference also shows how spatial frequency affects curvature. For a perturbation with amplitude a and wavenumber k,

$$w(x) = a\sin(kx), \qquad w_{xx} = -ak^2\sin(kx), \qquad |w_{xx}|_{\max} = |a|k^2. \tag{28}$$

The factor k^2 amplifies the curvature associated with high-frequency displacement variations. Consequently, a small displacement perturbation can produce appreciable bending-stress error in this reference problem. This observation concerns the curvature dependence of linear bending stress; it does not establish a k^2 law for von Mises stress in the automotive shell simulations.

B.4 Geometric Dependencies Beyond the Small-Deflection Assumption

The preceding analysis links positional geometry to global response and neighborhood geometry to local strain. When transverse deflection induces appreciable mid-surface stretching, pure bending is insufficient to describe the strains. We examine these dependencies using kinematics with small strains, moderate rotations, and quadratic transverse-gradient terms.

For the same initially flat, homogeneous, isotropic, and linear-elastic plate, we adopt the von Kármán approximation (Bower, 2025, Section 10.6.3), neglecting higher-order terms in in-plane displacement gradients and using linearized curvature. Let $\mathbf{u}_\parallel = (u, v)^\mathsf{T}$ be the in-plane mid-surface displacement. The in-plane strain tensor at distance ζ from the mid-surface is

$$\boldsymbol{\varepsilon} = \underbrace{\operatorname{sym}\nabla\mathbf{u}_\parallel}_{\text{in-plane displacement gradient}} + \underbrace{\frac{1}{2}\nabla w \otimes \nabla w}_{\substack{\text{deflection-induced}\\ \text{nonlinear stretching}}} - \underbrace{\zeta\mathbf{H}_w}_{\text{bending}}, \qquad \operatorname{sym}\mathbf{A} = \frac{1}{2}(\mathbf{A} + \mathbf{A}^\mathsf{T}). \tag{29}$$

All gradients are with respect to (x, y), and the off-diagonal tensor entries equal half the engineering shear strain γ_{xy}. The bending term $-\zeta\mathbf{H}_w$ is unchanged, so neighborhood direction and spacing retain their roles in representing local curvature through Eq. (27). Mid-surface stretching adds the in-plane displacement gradient and $\frac{1}{2}\nabla w \otimes \nabla w$. At $\zeta = 0$, these terms give

$$\varepsilon_x^0 = u_x + \frac{1}{2}w_x^2, \qquad \varepsilon_y^0 = v_y + \frac{1}{2}w_y^2, \qquad \gamma_{xy}^0 = u_y + v_x + w_x w_y. \tag{30}$$

The squared gradients w_x^2, w_y^2 contribute to normal strain, and $w_x w_y$ contributes to shear strain. These terms represent extension of mid-surface line elements caused by transverse deflection, so their evaluation also requires directions and physical scales.

To express this dependence through the same neighborhood geometry, write $\mathbf{d}_{ij} = \ell_{ij}\mathbf{n}_{ij}$. For locally smooth fields and sufficiently small neighborhoods, first-order expansions give

$$\begin{aligned} w_j - w_i &= \ell_{ij}\nabla w_i^\mathsf{T}\mathbf{n}_{ij} + O(\ell_{ij}^2), \\ \mathbf{u}_{\parallel,j} - \mathbf{u}_{\parallel,i} &= \ell_{ij}(\nabla\mathbf{u}_{\parallel,i})\mathbf{n}_{ij} + O(\ell_{ij}^2). \end{aligned} \tag{31}$$

Edge length converts nodal differences to variation per unit physical distance, and edge direction determines the observed gradient projection. In particular, the edge-direction component of the nonlinear strain induced by the transverse gradient satisfies

$$\mathbf{n}_{ij}^\mathsf{T}\left(\frac{1}{2}\nabla w_i \otimes \nabla w_i\right)\mathbf{n}_{ij} = \frac{1}{2}\left(\nabla w_i^\mathsf{T}\mathbf{n}_{ij}\right)^2 \approx \frac{1}{2}\left(\frac{w_j - w_i}{\ell_{ij}}\right)^2. \tag{32}$$

A fixed transverse displacement difference therefore corresponds to different stretching strains at different separations, while differently oriented edges measure different gradient projections. Edge directions and lengths support the interpretation of in-plane gradients and nonlinear transverse-gradient terms as well as curvature. Within the stated approximation, their relevance extends from pure bending to combined bending and mid-surface stretching.

At the structural level, mid-surface stretching feeds back into transverse motion through membrane forces, coupling local strains with overall deflection. Under the additional assumptions of no distributed in-plane forces and negligible in-plane inertia, while retaining transverse inertia, introduce an Airy membrane-force function Φ with $N_x = \Phi_{yy}$, $N_y = \Phi_{xx}$, and $N_{xy} = -\Phi_{xy}$. In-plane compatibility and transverse motion can then be written in the Föppl–von Kármán form:

$$\rho h w_{tt} + D\nabla^4 w - [\Phi, w] = q_c(\mathbf{x}, t), \qquad \nabla^4 \Phi + \frac{E_{\mathrm{p}} h}{2}[w, w] = 0, \tag{33}$$

where

$$[f, g] = f_{xx} g_{yy} + f_{yy} g_{xx} - 2 f_{xy} g_{xy}. \tag{34}$$

The bracket $[w, w]$ supplies the deflection-induced source in membrane-strain compatibility, while $[\Phi, w]$ represents the feedback of membrane forces on transverse motion. These terms couple local derivatives to the response over the finite domain. The membrane-force field depends on in-plane boundary conditions, and transverse motion remains constrained by the plate boundaries and driven by $\kappa(\mathbf{x} - \mathbf{p}; a(t))$ in Eq. (12). Identical impact-relative positions can consequently yield different responses in different structural regions.

Nonlinear coupling prevents the independent modal superposition in Eq. (18) from serving as a solution to this system. The response still depends on the positions of the load and finite boundaries. Thus, under small strains and moderate rotations, including mid-surface stretching preserves both geometric considerations: load-relative and structural positions govern the spatial setting, while neighborhood directions and scales enter local bending and stretching strains.

B.5 Connection to Our Geometric Design

The reference problems motivate two complementary uses of geometry. In global routing, node-to-slice assignments receive the impact-relative vector, its three-dimensional and XY-projected distances, and bounding-box positions. The relative features describe the node's relationship to the load; the box features provide a coarse location within the finite structure. Encoding these features adds a residual to the assignment logits, so identical impact-relative offsets in different structural regions can receive different weights. Bounding-box positions are inexpensive structural references, but do not measure distance to the actual boundary or fully describe the constraints.

In local propagation, MGN uses FE connectivity together with three-dimensional edge vectors and lengths. These inputs provide direction and scale when interpreting neighborhood feature differences, as motivated by the curvature, in-plane-gradient, and nonlinear stretching terms above. The analysis motivates the feature categories and their roles; experiments evaluate the particular descriptor and network implementation used here.

C GeoSlice-MGN Implementation Details

We specify how the geometric inputs enter the network in Section 3, from coordinate preprocessing and feature construction through local propagation and slice interaction to decoding. We also give the training objectives and panel-specific configurations.

C.1 Inputs and Geometric Preprocessing

Each node receives normalized coordinates together with the impact conditions:

$$\mathbf{f}_i = \left[\widetilde{\mathbf{x}}_i,\, \widetilde{\mathbf{p}},\, \widetilde{\mathbf{v}},\, \widetilde{\mu},\, \widetilde{E},\, \widetilde{\nu}\right] \in \mathbb{R}^{12} \tag{35}$$

Node and impact positions use the same coordinate transformation described below. Velocity, the mass scaling factor, and material parameters are standardized using training-set statistics.

Coordinate preprocessing. For the initial mesh, let $\bar{x}_a, s_a$ be the mean and standard deviation of node coordinates along axis $a \in \{x, y, z\}$. We apply the numerical floor used in the implementation to s_a and use

$$x'_{i,a} = \frac{x_{i,a} - \bar{x}_a}{s_a}, \qquad p'_a = \frac{p_a - \bar{x}_a}{s_a}, \qquad \Delta'_{i,a} = x'_{i,a} - p'_a. \tag{36}$$

Here, $\mathbf{x}'_i = \widetilde{\mathbf{x}}_i$ and $\mathbf{p}' = \widetilde{\mathbf{p}}$. Applying the same transformation to both preserves their relative offsets in normalized coordinates. Axis-specific scaling makes the resulting distances dimensionless and dependent on the scale of each coordinate axis. These network features therefore differ from the physical distances used in the mechanical analysis of Appendix B.

First five dimensions: impact-relative geometry.

$$\mathbf{g}_i^{\text{impact}} = [\Delta'_{i,x}, \Delta'_{i,y}, \Delta'_{i,z}, \|\mathbf{\Delta}'_i\|_2, \sqrt{(\Delta'_{i,x})^2 + (\Delta'_{i,y})^2}]. \tag{37}$$

The coordinate differences encode the direction and offset from impact. The other terms encode three-dimensional and XY-projected proximity. The z difference distinguishes shell elevations.

Last five dimensions: bounding-box position. We compute $x'_{\min}, x'_{\max}, y'_{\min}, y'_{\max}$ from the normalized initial mesh and define

$$\mathbf{g}_i^{\text{box}} = [x'_i - x'_{\min},\ y'_i - y'_{\min},\ x'_{\max} - x'_i,\ y'_{\max} - y'_i,\ q_i^x], \tag{38}$$

$$q_i^x = \text{clip}\left(\frac{x'_i - x'_{\min}}{\max(x'_{\max} - x'_{\min}, 10^{-6})}, 0, 1\right). \tag{39}$$

The first four entries measure offsets to the sides of the XY bounding box; q_i^x gives longitudinal position relative to its span. This encoding provides a structural position reference despite its algebraic redundancy. It uses the fixed vehicle coordinate system and therefore retains the distinction between longitudinal and lateral directions.

Table 6: Ten-dimensional descriptor for impact-conditioned routing, computed from normalized node and impact coordinates.

Feature group	Dim.	Meaning
Impact-relative coordinates	3	Node coordinates minus impact coordinates
3D/planar distances	2	Relative distances in 3D and the XY plane
Lower bounding-box offsets	2	Offsets from the minimum XY coordinates
Upper bounding-box offsets	2	Offsets to the maximum XY coordinates
Longitudinal box coordinate	1	Normalized x position q_i^x

C.2 Local Propagation and Slice Interaction

Local graph propagation. The edge input uses the four-dimensional features in Eq. (3). Node encoder E_v and edge encoder E_e map their respective inputs into a $C = 160$-dimensional latent space:

$$\mathbf{h}_i^{(0)} = E_v(\mathbf{f}_i), \qquad \mathbf{e}_{ij}^{(0)} = E_e(\mathbf{f}_{ij}). \tag{40}$$

Local encoding and post-deslicing refinement use the same form of MGN message passing. Each block ℓ updates edge states, averages incoming edge messages, and applies a residual node update:

$$\Delta\mathbf{e}_{ij}^{(\ell)} = \phi_e^{(\ell)}\left(\mathbf{h}_i^{(\ell)}, \mathbf{h}_j^{(\ell)}, \mathbf{e}_{ij}^{(\ell)}\right), \tag{41}$$

$$\mathbf{e}_{ij}^{(\ell+1)} = \text{LN}_e\left[\mathbf{e}_{ij}^{(\ell)} + \Delta\mathbf{e}_{ij}^{(\ell)}\right], \tag{42}$$

$$\mathbf{m}_i^{(\ell+1)} = \text{Mean}_{j:(j,i)\in\mathcal{A}}\, \mathbf{e}_{ji}^{(\ell+1)}, \tag{43}$$

$$\mathbf{h}_i^{(\ell+1)} = \text{LN}_v\left[\mathbf{h}_i^{(\ell)} + \phi_v^{(\ell)}\left(\mathbf{h}_i^{(\ell)}, \mathbf{m}_i^{(\ell+1)}, \mathbf{f}_i\right)\right]. \tag{44}$$

Both $\phi_e^{(\ell)}$ and $\phi_v^{(\ell)}$ are MLPs, and every node update also receives the original features $\mathbf{f}_i$. The first three blocks evolve the encoded edge states continuously. After deslicing, the two refinement blocks use the resulting edge states to continue the edge-state computation across the global module.

Impact-conditioned routing. The encoder f_θ maps the ten-dimensional descriptor to one residual per head–slice pair: 4×48 values for the two floorfront panels and 4×96 for `trunkfloor`. Its output layer is zero-initialized. A learnable scalar α_h in each head is passed through a sigmoid to obtain γ_h, which gates the residual before addition to the content logits in Eq. (7).

A linear projection of $\mathrm{LN}(\mathbf{h}_i)$ produces content logits, and a separate projection produces node values. The assignment, head-wise value, and aggregated-state tensors have shapes $N_v \times H_\mathrm{s} \times S$, $N_v \times H_\mathrm{s} \times d$, and $H_\mathrm{s} \times S \times d$, respectively. Here, $H_\mathrm{s} = 4$ and $d = C/H_\mathrm{s} = 40$.

Occupancy representation and slice interaction. Normalized aggregation records average features but does not separately retain assignment mass. For example, if all node features are identical and the stabilization term is ignored, every normalized slice state equals that feature vector regardless of its total assignment weight. The occupancy residual records this information through a clipped relative log-occupancy:

$$r_{hs} = \mathrm{clip}\left(\log\left[\max\left(\frac{\mathrm{sg}(m_{hs})}{N_v/S}, 10^{-6}\right)\right], -6, 6\right), \tag{45}$$

Here, sg denotes stop-gradient. Positive and negative values indicate occupancy above and below the uniform reference. Detachment applies only to this residual path; assignment weights continue to receive gradients through ordinary aggregation. Slice construction follows Eqs. (8)–(9), with a zero-initialized output layer in the shared map ϕ.

Four Transformer blocks then apply attention across slices within each head. QKV projections are shared across heads; the output projection and FFN mix channels across heads at each slice index.

C.3 Joint-Field Decoding and Model Configurations

After global interaction, Eq. (10) maps slice states back to nodes using A_{ihs}. We concatenate the head-wise outputs, project them to C channels, and add them to the local node representations through a residual connection. Two MGN refinement blocks produce the final states $\mathbf{h}_i^{\mathrm{ref}}$.

The displacement head takes $[\mathbf{h}_i^{\mathrm{ref}}, \mathbf{f}_i]$ as input:

$$\widehat{\mathbf{u}}_i^{\star} = D_u\big([\mathbf{h}_i^{\mathrm{ref}}, \mathbf{f}_i]\big) \in \mathbb{R}^3. \tag{46}$$

For the vertex set $\mathcal{V}(e)$ of shell element e, we first compute

$$\mathbf{h}_e^{\mathrm{ref}} = \frac{1}{|\mathcal{V}(e)|} \sum_{i \in \mathcal{V}(e)} \mathbf{h}_i^{\mathrm{ref}}, \qquad \mathbf{f}_e = \frac{1}{|\mathcal{V}(e)|} \sum_{i \in \mathcal{V}(e)} \mathbf{f}_i, \tag{47}$$

and predict normalized effective stress with $D_\sigma([\mathbf{h}_e^{\mathrm{ref}}, \mathbf{f}_e])$.

Table 7 summarizes the model configurations for the three panels.

Table 7: GeoSlice-MGN configurations for the three panels.

Configuration	`floorfrontdriver`	`floorfrontR`	`trunkfloor`
Node/edge input dimensions	12/4	12/4	12/4
Hidden dimension C	160	160	160
Local encoding MGN blocks	3	3	3
Slicing heads H_s	4	4	4
Slices per head S	48	48	96
Head dimension d	40	40	40
Slice Transformer blocks	4	4	4
Refinement MGN blocks	2	2	2

Target normalization and training loss. We normalize each displacement component and stress by their respective training-set RMS scales. After inference, we use the same scales to restore physical

units. Independent models use statistics from their own panel's training set; shared models use the joint training set. Independent training gives equal weight to normalized displacement and stress MSE:

$$\mathcal{L}_U = \frac{1}{3N_v} \sum_{i=1}^{N_v} \|\widehat{\widetilde{\mathbf{u}}}_i^\star - \overline{\mathbf{u}}_i^\star\|_2^2, \tag{48}$$

$$\mathcal{L}_\sigma = \frac{1}{N_e} \sum_{e} \left(\widehat{\widetilde{\sigma}}_{e,\mathrm{eff}}^\star - \overline{\sigma}_{e,\mathrm{eff}}^\star \right)^2, \qquad \mathcal{L} = \mathcal{L}_U + \mathcal{L}_\sigma. \tag{49}$$

Fixed constrained nodes are included in the displacement loss. Nodes and elements receive no additional spatial weighting.

Shared multi-panel modeling. For each shared-model update, we sample one case per panel and average the displacement and stress relative L_2 losses computed in physical units, with equal weights across tasks and panels. Shared GeoSlice-MGN uses 96 slices per head and otherwise retains the local–global–local backbone. Normalization uses only joint training data, and checkpoint selection uses validation macro-average error. At test time, cases are equally weighted within each panel. Table 3 reports each panel's mean and sample standard deviation across three runs.

D Baseline Models and Control Configurations

We describe the baseline task interfaces and architectures on `AutoPanelImpact` and the controls for geometry injection and all-node attention.

D.1 Task Interfaces and Configuration Overview

We compare GeoSlice-MGN against seven baselines: POD+MLP, DeepONet, MGN, GNO, GINO, Geo-FNO, and Transolver. Each method retains its core modeling mechanism and jointly predicts peak displacement and stress in one forward pass. DeepONet uses its branch–trunk interface; the other neural networks use the 12-dimensional node input in Eq. (35). MGN and GNO additionally receive the edge features in Eq. (3).

All models predict three-dimensional displacement at FE nodes. DeepONet queries stress at element centers. The other neural networks average final node representations over element vertices and apply an element decoder shared across the mesh. These interfaces use consistent supervision and evaluation locations across methods.

On `AutoPanelImpact`, neural baselines and control models use approximately matched parameter budgets of around 2.5 million trainable parameters. The configuration of GeoSlice-MGN for each panel is given in Table 7.

D.2 Baseline Implementations

POD+MLP. POD+MLP follows non-intrusive reduced-order modeling: POD extracts spatial bases from response snapshots, and an MLP maps input parameters to basis coefficients (Hesthaven & Ubbiali, 2018). For each panel, we normalize and center training displacement and stress separately, then construct one POD basis for each field. We map nine standardized conditioning variables (impact location, velocity, mass scaling, and material parameters) to two coefficient sets. Each field is reconstructed from its basis combination and training mean.

DeepONet. DeepONet combines branch features for an input function or condition with trunk features for a query location through an inner product (Lu et al., 2021). Here, the branch receives normalized impact position and six standardized physical conditions, while the trunk receives three-dimensional coordinates. The shared trunk is queried at nodes for three displacement channels and at shell-element centers for effective stress. This gives a low-rank baseline that uses coordinate queries without FE edge connectivity.

MGN. MGN encodes mesh nodes and edges, propagates their hidden states through local message passing, and decodes physical responses (Pfaff et al., 2021). Our adaptation uses separate node and edge encoders with alternating residual edge and node updates on first-order FE adjacency. Final node states predict peak displacement; their element-wise averages feed the stress decoder. The resulting baseline represents both fields using local mesh propagation.

GNO. GNO approximates function mappings by discretizing learnable kernel integrals over graph neighborhoods (Li et al., 2020; Kovachki et al., 2023). We use the same first-order FE adjacency as MGN. Each block computes kernel weights from source and target features and relative edge geometry, applies them to source features, and updates nodes through neighborhood-mean integration and a residual connection. Node and element decoders then produce displacement and stress. This compares kernel-weighted integration with edge-state message passing on the same neighborhoods.

GINO. GINO uses graph integral transforms to connect irregular input geometries with a regular latent grid, where an FNO performs global spectral interaction (Li et al., 2023b). For shell meshes, normalized XY coordinates define the two-dimensional latent grid and integration neighborhoods. Node features are integrated onto the grid, processed spectrally, and queried back at FE nodes before joint-field decoding. Coupling uses coordinate neighborhoods without FE edge connectivity.

Geo-FNO. Geo-FNO learns a map from physical geometry to a regular latent domain for FFT-based spectral processing (Li et al., 2023a). Our adaptation maps FE nodes to the unit square through a node-conditioned deformation network, deposits their features on a regular two-dimensional grid by bilinear splatting, and applies FNO layers. Bilinear sampling returns the processed features to the original nodes for joint-field decoding. This preserves the combination of learned geometric mapping and latent spectral interaction on shell meshes.

Transolver. Transolver's Physics-Attention softly aggregates mesh points into learnable slices, applies attention between slice states, and reuses the assignments to return updates to the mesh (Wu et al., 2024). We stack these blocks and attach node and element decoders for peak displacement and stress. Assignments depend on node hidden states, with no FE edge connectivity or additional impact-relative bias. This is a slice-attention baseline for global interaction.

D.3 Geometry and All-Node Attention Controls

MGN+Geo10. Concatenating the ten-dimensional descriptor expands the MGN node input from 12 to 22 dimensions. Backbone width and depth are retained, with no routing bias or occupancy branch.

Transolver+GeoRoute10. This control adds a gated bias from the ten-dimensional descriptor to Transolver's assignment scores. It retains the backbone and input/output interfaces and introduces no occupancy branch.

MGN+FullAttention. This control places four all-node attention blocks between three local encoding MGN blocks and two local refinement MGN blocks.

E Public Dataset Adaptations and Experimental Settings

The two public tasks use different prediction interfaces: peak displacement and stress for Bumper-Beam, and a complete deformation sequence for Plasticity. This section gives the preprocessing steps, model adaptations, and training protocols for each task. Test results appear in Table 4.

E.1 BUMPERBEAM

E.1.1 DATA AND PREDICTION TARGETS

BumperBeamCrashExample is a public automotive crash simulation dataset based on OpenRadioss and generated by the OpenRadioss2PhysicsNeMo project[1]. The dataset contains 131 independent simulations of a moving bumper contacting a fixed rigid cylinder. The reference mesh and initial velocity are fixed, while shell thickness and lateral cylinder position vary. Thickness ranges from 1.21 to 2.00 mm, cylinder-axis position from -480.26 to 491.83 mm, and initial velocity is -5 m/s along x. The prediction task maps the initial structure and collision conditions to three-dimensional displacement and shell-element effective stress at peak deformation.

All cases use 13,676 nodes and 13,678 quadrilateral shell elements. Bidirectional perimeter connectivity gives 54,704 deduplicated directed edges. We use the 11 states at $0, 10, \ldots, 100$ ms and select the state with the largest displacement magnitude over valid nodes, taking the earliest state in a tie. Displacement and stress targets come from this same state. Peak time is used only for target construction and is excluded from model inputs.

The 108 nodes associated with the rigid-body definition remain in the graph but are omitted from displacement supervision and evaluation, leaving 13,568 valid nodes. Stress is evaluated on every shell element. Displacement is in mm, and the released `cell_stress_vm` values are converted from GPa to MPa. We split complete cases into 91 training, 20 validation, and 20 test cases, keeping every state of a case in the same subset.

E.1.2 INPUTS AND MODEL ADAPTATIONS

Each node receives eight inputs: its coordinates, a cylinder-axis reference position, shell thickness, and velocity. The reference $(-170, y_{\text{pole}}, 0)$ mm supplies a known loading location. Node and reference positions share coordinate normalization; thickness and velocity use training-set standardization. GeoSlice-MGN has hidden dimension 160, with 3/4/2 layers for local encoding, slice interaction, and refinement, and four routing heads with 96 slices each. The ten-dimensional branch uses the cylinder-axis reference to construct relative geometry, and the occupancy residual uses stop-gradient.

Transolver follows the official irregular-mesh implementation with six layers, width 188, four heads, and 64 slices. It averages nodal stress predictions over element vertices; the other models use their element decoders. MGN and GNO use four layers of widths 236 and 211, respectively, for message passing and kernel integration along FE adjacency.

Geo-FNO and GINO use our two-dimensional latent-grid adaptations. Geo-FNO combines learned two-dimensional coordinates, grid splatting, and resampling. GINO retains three-dimensional node features but defines integration neighborhoods in normalized XY coordinates. Both use a 96×48 grid and four FNO layers. Their hidden widths are 60 and 92 and their spectral modes are 6×7 and 4×4, respectively. Geo-FNO's deformation network has width 73. GINO uses radius 0.35 and retains 8–32 nearest neighbors per point.

E.1.3 TRAINING AND EVALUATION

Displacement and stress are separately scaled by training-set RMS values. Training minimizes an equally weighted sum of valid-node three-component displacement MSE and element stress MSE. Checkpoint selection minimizes the equally weighted mean of validation displacement and stress relative L_2 errors. Both test metrics use the selected checkpoint. For Table 4, we first average errors equally across 20 test cases in each run, then report the mean and sample standard deviation across three runs.

[1]Data source: AIRBORNEPANDA/BumperBeamCrashExample (pinned version); generation project: OpenRadioss2PhysicsNeMo.

E.2 PLASTICITY

E.2.1 DATA AND PREDICTION TARGETS

Each case pairs a die profile with the induced deformation sequence: deformed coordinates and displacements on a 101×31 grid over 20 frames. The official pool of 900 training cases is split into fixed subsets of 810 training and 90 validation cases. All frames from a case remain together, and the results in Table 4 use validation-selected checkpoints evaluated on 80 separate held-out test cases.

Die profiles use training-set standardization; output values retain their original scales. Bidirectional row and column connections define a reference graph with 3131 nodes and 12260 directed edges.

E.2.2 INPUTS AND MODEL ADAPTATIONS

Task-condition representation. For die indentation, the die profile supplies the loading geometry, while logical coordinates and boundary indicators locate each response point in the reference domain. We combine them in the seven-dimensional descriptor

$$\mathbf{g}_i^{\mathrm{P}} = [\xi_i, \eta_i, \widetilde{h}(\xi_i), b_i^{\xi-}, b_i^{\xi+}, b_i^{\eta-}, b_i^{\eta+}] \in \mathbb{R}^7. \tag{50}$$

Here, $(\xi_i, \eta_i) \in [0,1]^2$ are logical coordinates. At horizontal position ξ_i, the standardized profile value $\widetilde{h}(\xi_i)$ is broadcast along η, preserving its correspondence with the grid. The four binary indicators mark $\xi = 0$, $\xi = 1$, $\eta = 0$, and $\eta = 1$. They distinguish domain edges from the interior, without assigning contact or constraint types. Logical-coordinate differences and their lengths define local edge geometry.

Both the node encoder and routing branch receive this task-specific descriptor. Table 4 therefore reports sequence-prediction results for the adapted full model. The thin-plate analysis in Section 3.2 motivates geometry for the impact task, whose design is tested separately by the ablations and routing diagnostics in Section 4.4.

Full-sequence prediction. Each model predicts all 20 frames in one forward pass from die and mesh information. The four channels (x, y, u_x, u_y) form one full-sequence target per case. GeoSlice-MGN retains hidden dimension 160, with 3, 4, and 2 layers for local encoding, slice processing, and refinement; 4 heads with 48 slices each; and FFN width 648. Routing adds a gated residual from $\mathbf{g}_i^{\mathrm{P}}$ to content logits, and the occupancy path uses stop-gradient. The decoder concatenates hidden states with the original inputs to predict 80 channels, which it reshapes into the sequence.

Transolver uses its Structured Mesh 2D backbone: eight layers, width 128, eight heads, and 64 slices. Convolutional projections and learnable temperature are retained, while outputs are adapted to the same 80-channel representation. GeoSlice-MGN and Transolver have 2,436,460 and 2,822,160 parameters, respectively.

Other baseline configurations. Geo-FNO uses four FNO3d layers with width 8, modes $(11, 12, 9)$, padding 5, and projection width 236. Counting complex weights by real-valued degrees of freedom gives 2,436,456 parameters. MGN uses 15 layers, width 128, and mean aggregation, totaling 2,356,816 parameters.

GINO integrates inputs onto a 64×32 grid with radius 0.05, applies four FNO layers of width 64 with 16×8 modes, and maps the result back to the original grid. It has 8,574,032 parameters. GNO uses four channel-diagonal linear kernel layers of width 64 and radius 0.1, totaling 335,056 parameters. Inputs, outputs, and splits are shared across baselines.

E.2.3 TRAINING AND EVALUATION

All models are trained using the joint relative error over the full sequence:

$$\mathcal{L}_{\mathrm{seq}} = \frac{\|\widehat{Y} - Y\|_F}{\max(\|Y\|_F, 10^{-8})}, \tag{51}$$

The norm includes every node, frame, and output channel. Cases receive equal weights within each batch, and we select checkpoints by the lowest validation joint error. Each model is trained three

times; test errors are averaged equally over cases before computing the mean and sample standard deviation across runs. The joint relative L_2 includes all four channels, while displacement relative L_2 includes only (u_x, u_y). Table 4 reports both metrics.

F Supplementary Ablation Studies

These experiments examine where geometry is injected, the placement of local layers around global interaction, and the contributions of individual descriptor components. Unless stated otherwise, each configuration is trained independently on `floorfrontdriver`, with checkpoints selected by validation error and evaluated on the same test cases. Tables report physical-field relative L_2 errors as means and sample standard deviations across three runs.

F.1 Geometric Injection Location

We compare three injection locations while retaining the 12-dimensional node input, ten-dimensional descriptor, occupancy representation, and local–global backbone. The geometric MLP has dimensions $10 \to 160 \to 160 \to (4 \times 48)$ with head-wise gating in every variant; other settings are fixed.

Post-MGN node state applies a fixed mapping to obtain a 160-dimensional geometric residual and adds it after the first three MGN layers. Geometry then influences both assignments and slice contents. *Decoder state* adds that residual after the two refinement layers, affecting displacement decoding and the node aggregation used for stress decoding. *Assignment logits* adds the geometric output directly before the assignment softmax, so geometry enters at the routing stage.

Table 8: Test relative L_2 errors for different geometric injection locations (mean±sample standard deviation across three independent runs).

Configuration	Displacement e_U	Stress e_σ
Decoder state	0.1791 ± 0.0126	0.1965 ± 0.0049
Post-MGN node state	0.1651 ± 0.0062	0.1916 ± 0.0042
Assignment logits	$\mathbf{0.1506 \pm 0.0049}$	$\mathbf{0.1825 \pm 0.0042}$

Assignment-logit injection yields the lowest mean errors for both fields in Table 8. Earlier injection into node states also improves on decoder-only injection, but remains less accurate than direct routing modulation. In this comparison, using geometry to organize slice assignments is more effective than injecting it at either tested node-state location.

F.2 Local–Global–Local Layout

We next vary the placement of local propagation while keeping the total depth at five MGN layers and four slice-processing layers. Hidden dimension, slice count, and overall model scale are shared.

The "3 pre + slicing + 2 post" layout encodes neighborhoods before slicing and refines the returned global information after deslicing. The "5 pre + slicing + 0 post" layout places every MGN layer before global interaction and decodes immediately afterward. This comparison assesses local refinement while keeping the overall network scale essentially unchanged.

Table 9: Test relative L_2 errors for different local–global layouts (mean±sample standard deviation across three independent runs).

Configuration	Displacement e_U	Stress e_σ
5 pre + slicing + 0 post	0.1781 ± 0.0037	0.2048 ± 0.0002
3 pre + slicing + 2 post	$\mathbf{0.1506 \pm 0.0049}$	$\mathbf{0.1825 \pm 0.0042}$

The "3 pre + slicing + 2 post" layout gives lower mean errors for both fields in Table 9. For these configurations, placing all local layers before global interaction is less effective than retaining

post-deslicing refinement. This result supports refining the returned global features within FE neighborhoods before decoding.

F.3 Geometric Descriptor Composition

Finally, we evaluate the two feature groups in the descriptor separately. Five features describe three-dimensional impact-relative offsets and the corresponding three-dimensional and planar distances; five describe position relative to the XY bounding box. *Impact-only* retains the first group, and *Boundary-only* retains the box group. The latter name refers to the bounding-box features, which provide a coarse structural reference and do not measure distances to actual boundaries. *No explicit geometry* removes the routing descriptor while retaining occupancy and is the direct control for the geometric branch.

Table 10: Test relative L_2 errors for different geometric descriptor compositions (mean$\pm$sample standard deviation across three independent runs).

Configuration	Displacement e_U	Stress e_σ
No explicit geometry	0.1648 ± 0.0059	0.1876 ± 0.0041
Boundary-only	0.1607 ± 0.0013	0.1863 ± 0.0045
Impact-only	0.1578 ± 0.0026	0.1836 ± 0.0044
Full 10D geometry	$\mathbf{0.1506 \pm 0.0049}$	$\mathbf{0.1825 \pm 0.0042}$

Both feature groups lower the mean errors relative to *No explicit geometry* in Table 10, with larger reductions from impact-relative features. Combining them gives the lowest mean displacement and stress errors. These results support using structural position as a complement to impact-relative geometry in the tested routing configuration.

G Spatial Error Analysis

Full-field metrics summarize prediction errors over the entire mesh. We locate errors by evaluating the impact neighborhood separately and examining their variation with impact distance.

G.1 Prediction Errors in the Impact Neighborhood

Localized loading produces steep response gradients in the impact neighborhood. Using initial node coordinates $\mathbf{x}_i$ and impact location $\mathbf{p}$, define the normalized distance

$$d_{\max} = \max_{j \in \mathcal{V}} \|\mathbf{x}_j - \mathbf{p}\|_2, \qquad \rho(\mathbf{x}) = \frac{\|\mathbf{x} - \mathbf{p}\|_2}{d_{\max}}. \tag{52}$$

Node distances use $\mathbf{x}_i$; element distances use $\mathbf{c}_e$, the mean position of their valid nodes. With $\alpha = 0.2$, the evaluation regions are

$$\mathcal{V}_\alpha = \{i : \rho(\mathbf{x}_i) \le \alpha\}, \qquad \mathcal{E}_\alpha = \{e : \rho(\mathbf{c}_e) \le \alpha\}. \tag{53}$$

The reference mesh and impact position jointly determine these regions. Every model uses the same masks, whose spatial extent is illustrated in Fig. 5.

Local displacement and stress relative L_2 errors are computed over $\mathcal{V}_\alpha$ and $\mathcal{E}_\alpha$, respectively, and denoted by $e_{U,\alpha}$ and $e_{\sigma,\alpha}$. To express errors in physical units as well, let $\Delta\mathbf{u}_i = \widehat{\mathbf{u}}_i^\star - \mathbf{u}_i^\star$ and $\Delta\sigma_e = \widehat{\sigma}_{e,\text{eff}}^\star - \sigma_{e,\text{eff}}^\star$, and define

$$R_{U,\alpha} = \sqrt{\frac{1}{|\mathcal{V}_\alpha|} \sum_{i \in \mathcal{V}_\alpha} \|\Delta\mathbf{u}_i\|_2^2}, \qquad R_{\sigma,\alpha} = \sqrt{\frac{1}{|\mathcal{E}_\alpha|} \sum_{e \in \mathcal{E}_\alpha} (\Delta\sigma_e)^2}. \tag{54}$$

Displacement RMSE averages squared three-dimensional vector errors over nodes and is reported in mm; stress RMSE averages squared element errors and is reported in MPa. For Table 11, we weight cases equally within each run, then compute means and sample standard deviations across three runs.

GeoSlice-MGN achieves the lowest means on five of six regional relative L_2 metrics. It also has the lowest values for all relative-error and RMSE metrics on `floorfrontR` and `trunkfloor`. On `floorfrontdriver`, MGN+FullAttention has lower stress errors, and Transolver+GeoRoute10 has a slightly lower displacement RMSE. The full-field advantage thus extends to most local metrics, though the exceptions depend on the panel and target.

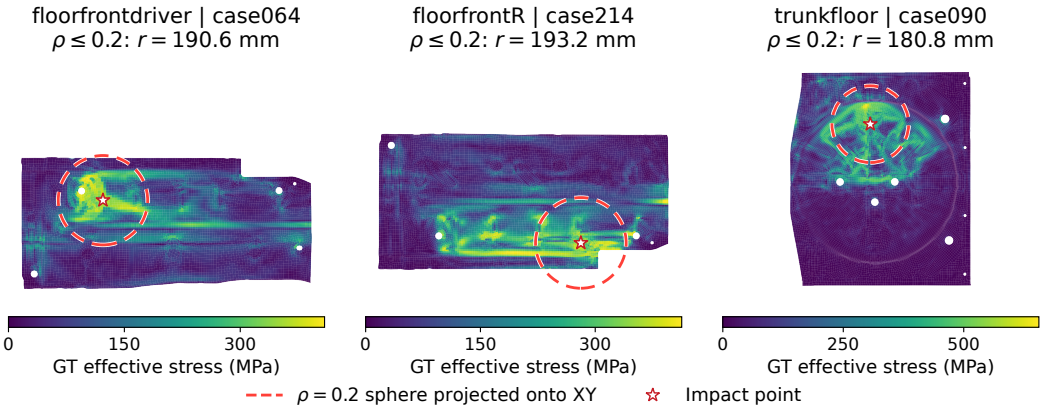


Figure 5: Ground-truth effective stress and impact neighborhoods $\rho \le 0.2$ on the three panels. Stars mark impact locations; dashed circles show the XY-plane projection of a three-dimensional distance sphere with radius $0.2d_{\max}$. Local errors are computed over nodes and element centers selected using three-dimensional distances. Fields are plotted on the reference mesh, with equal aspect ratios and independent color scales (MPa) for each panel. The circles mark the evaluation extent; the FE contact conditions determine the actual contact region.

Table 11: Test errors within the impact neighborhood $\rho \leq 0.2$. Metrics are computed within each case's neighborhood, then averaged equally over test cases. Values are mean±sample standard deviation across three independent runs. e_U, e_σ are local relative L_2 errors; R_U is the three-dimensional displacement-vector RMSE (mm), and R_σ is stress RMSE (MPa). Bold indicates the lowest mean for each panel and metric. † denotes a control constructed in this work.

Model	e_U	e_σ	R_U (mm)	R_σ (MPa)
		`floorfrontdriver`		
POD+MLP	0.1838 ± 0.0060	0.2282 ± 0.0043	1.444 ± 0.018	54.76 ± 1.03
DeepONet	0.1619 ± 0.0105	0.2109 ± 0.0056	1.279 ± 0.039	50.65 ± 1.24
GNO	0.1237 ± 0.0010	0.1702 ± 0.0066	1.010 ± 0.015	41.20 ± 1.60
GINO	0.1286 ± 0.0043	0.1683 ± 0.0004	1.060 ± 0.023	40.64 ± 0.07
Geo-FNO	0.1284 ± 0.0031	0.1955 ± 0.0039	1.024 ± 0.037	47.32 ± 1.02
MGN	0.1261 ± 0.0040	0.1819 ± 0.0043	1.035 ± 0.020	44.01 ± 1.05
MGN+Geo10†	0.1213 ± 0.0061	0.1816 ± 0.0017	0.988 ± 0.021	44.04 ± 0.42
Transolver	0.1114 ± 0.0014	0.1602 ± 0.0017	0.913 ± 0.012	38.70 ± 0.39
Transolver+GeoRoute10†	0.1046 ± 0.0036	0.1588 ± 0.0045	$\mathbf{0.851 \pm 0.024}$	38.43 ± 0.98
MGN+FullAttention†	0.1038 ± 0.0034	$\mathbf{0.1522 \pm 0.0002}$	0.854 ± 0.010	$\mathbf{36.76 \pm 0.06}$
GeoSlice-MGN	$\mathbf{0.1006 \pm 0.0036}$	0.1545 ± 0.0024	0.852 ± 0.029	37.37 ± 0.61
		`floorfrontR`		
POD+MLP	0.1471 ± 0.0048	0.2339 ± 0.0043	1.684 ± 0.034	55.65 ± 0.96
DeepONet	0.1265 ± 0.0042	0.2161 ± 0.0048	1.439 ± 0.058	51.11 ± 1.03
GNO	0.0991 ± 0.0043	0.1724 ± 0.0019	1.092 ± 0.023	41.18 ± 0.42
GINO	0.1157 ± 0.0049	0.1823 ± 0.0022	1.287 ± 0.073	43.20 ± 0.43
Geo-FNO	0.1066 ± 0.0061	0.2054 ± 0.0073	1.196 ± 0.081	49.07 ± 1.81
MGN	0.1023 ± 0.0050	0.1883 ± 0.0021	1.126 ± 0.044	44.96 ± 0.41
MGN+Geo10†	0.0978 ± 0.0049	0.1857 ± 0.0016	1.074 ± 0.049	44.38 ± 0.43
Transolver	0.0887 ± 0.0041	0.1638 ± 0.0041	0.985 ± 0.056	39.00 ± 1.01
Transolver+GeoRoute10†	0.0805 ± 0.0026	0.1619 ± 0.0036	0.897 ± 0.033	38.57 ± 0.92
MGN+FullAttention†	0.0896 ± 0.0009	0.1590 ± 0.0025	0.976 ± 0.015	37.74 ± 0.62
GeoSlice-MGN	$\mathbf{0.0783 \pm 0.0021}$	$\mathbf{0.1506 \pm 0.0031}$	$\mathbf{0.863 \pm 0.016}$	$\mathbf{35.90 \pm 0.68}$
		`trunkfloor`		
POD+MLP	0.4636 ± 0.0182	0.3412 ± 0.0011	3.046 ± 0.096	95.30 ± 0.05
DeepONet	0.4331 ± 0.0288	0.3608 ± 0.0062	2.855 ± 0.216	100.37 ± 1.54
GNO	0.3569 ± 0.0285	0.3411 ± 0.0070	2.301 ± 0.187	95.42 ± 1.79
GINO	0.3236 ± 0.0035	0.3090 ± 0.0080	2.015 ± 0.029	85.53 ± 2.11
Geo-FNO	0.3646 ± 0.0266	0.3357 ± 0.0070	2.219 ± 0.191	94.07 ± 2.11
MGN	0.3482 ± 0.0241	0.3394 ± 0.0051	2.213 ± 0.137	94.94 ± 1.43
MGN+Geo10†	0.3289 ± 0.0151	0.3381 ± 0.0091	2.108 ± 0.106	94.45 ± 2.17
Transolver	0.3290 ± 0.0320	0.3308 ± 0.0063	2.049 ± 0.153	92.41 ± 2.11
Transolver+GeoRoute10†	0.2740 ± 0.0107	0.3026 ± 0.0057	1.729 ± 0.072	84.60 ± 1.59
MGN+FullAttention†	0.3241 ± 0.0140	0.3248 ± 0.0023	2.053 ± 0.086	90.38 ± 0.51
GeoSlice-MGN	$\mathbf{0.2583 \pm 0.0120}$	$\mathbf{0.2773 \pm 0.0108}$	$\mathbf{1.674 \pm 0.091}$	$\mathbf{77.47 \pm 3.34}$

G.2 Error Variation with Impact Distance

To examine how error changes from the impact neighborhood to the far field, we partition ρ in Eq. (52) into 10 equal-width bins. For each case and bin, relative L_2 errors are computed for three-component displacement and element stress, then averaged equally over test cases. Figure 6 shows how the errors vary spatially.

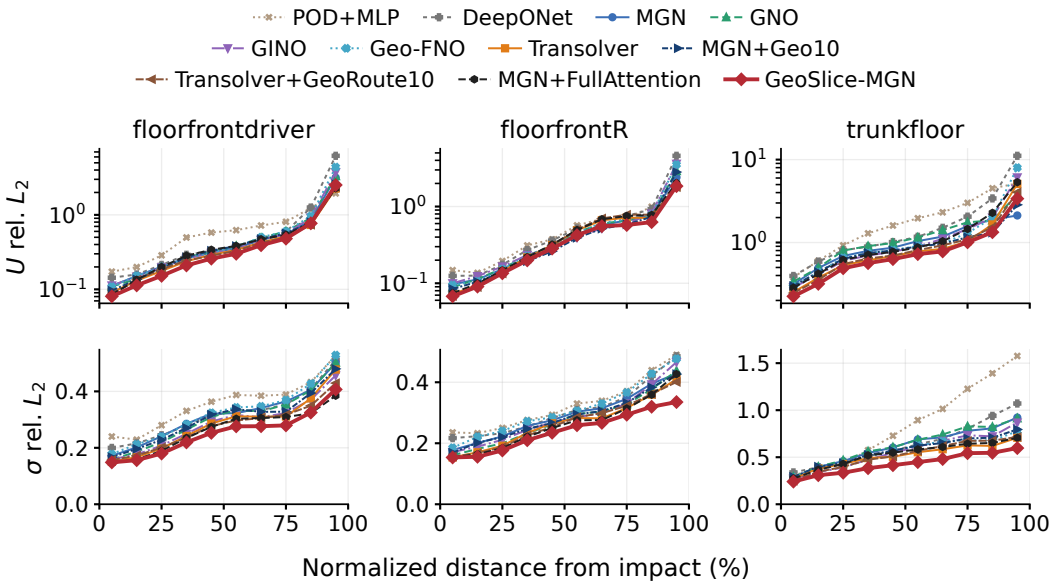


Figure 6: Error variation with normalized impact distance. Top: three-dimensional displacement. Bottom: effective stress. Points mark bin midpoints and show equally weighted means of within-bin relative L_2 errors across test cases. The displacement-error axis is logarithmic. Model configurations match Table 1.

Relative errors generally increase with impact distance, particularly for displacement in the farthest bins. Each value uses the ground-truth norm within its own bin as the denominator, so deviations in low-amplitude regions can produce large relative errors. The curves therefore reflect both the prediction error and the spatial variation in ground-truth magnitude; they do not represent absolute-error profiles.

H PREDICTION FIELDS AND ERROR VISUALIZATIONS

We visualize displacement magnitude and effective stress for `floorfrontdriver`, `floorfrontR`, and `trunkfloor`. We use model checkpoints trained with the same seed to predict response fields at the selected peak state $t^\star$ for each illustrated case. Each figure contains three comparison blocks, with response fields above their absolute errors. Displacement error in these plots is the absolute difference between magnitudes, distinct from the three-dimensional vector error used in the quantitative metrics.

For `floorfrontdriver`, Fig. 7 shows a central region of high displacement. GeoSlice-MGN reproduces its location, with errors concentrated around that region. MGN, GNO, and Transolver show broader bright error regions in the same area.

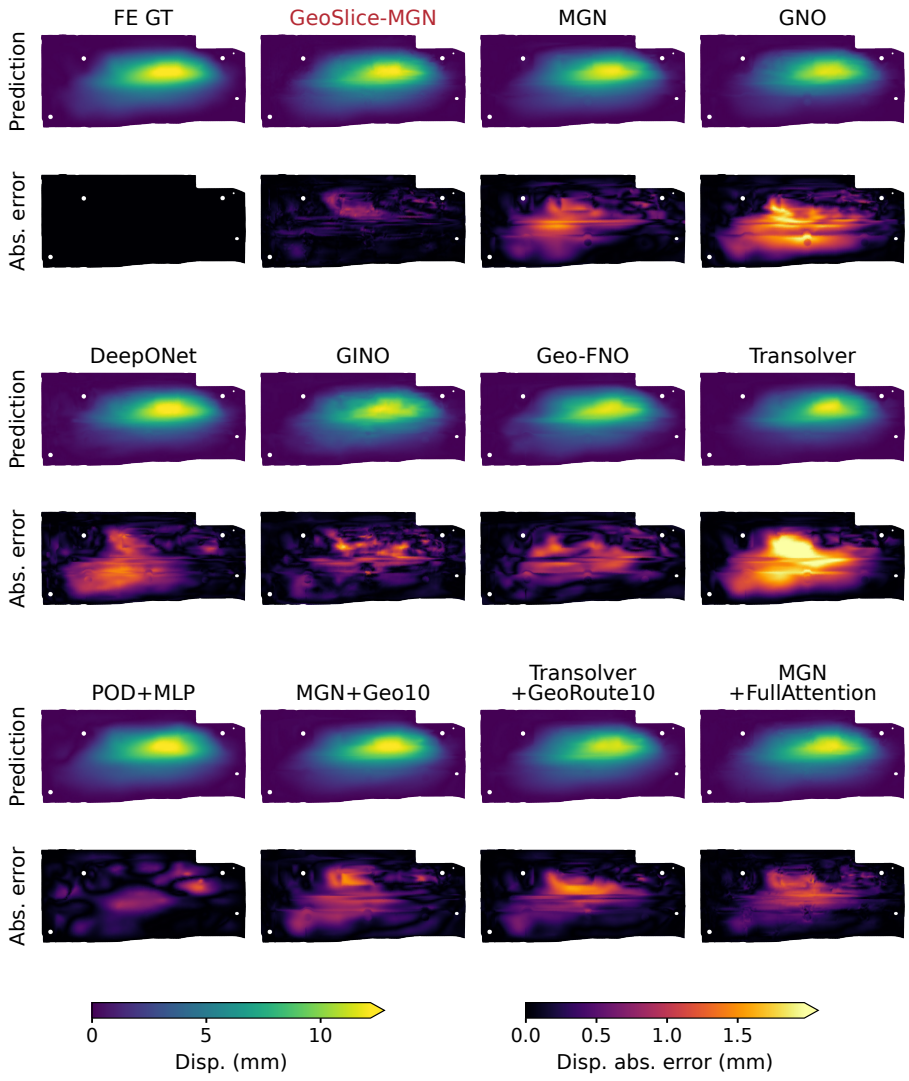


Figure 7: Nodal displacement magnitudes $\{\|\mathbf{u}_i^\star\|_2\}_{i=1}^{N_v}$ on `floorfrontdriver` and their node-wise absolute errors $\left|\|\widehat{\mathbf{u}}_i^\star\|_2 - \|\mathbf{u}_i^\star\|_2\right|$. FE GT has zero error by definition. The bottom-left color bar shows displacement magnitude and the bottom-right bar shows magnitude error, both in mm.

Stress on the same panel forms bands and localized patches, especially on the right (Fig. 8). GeoSlice-MGN reproduces the main patterns, while its remaining errors are concentrated around these high-stress regions.

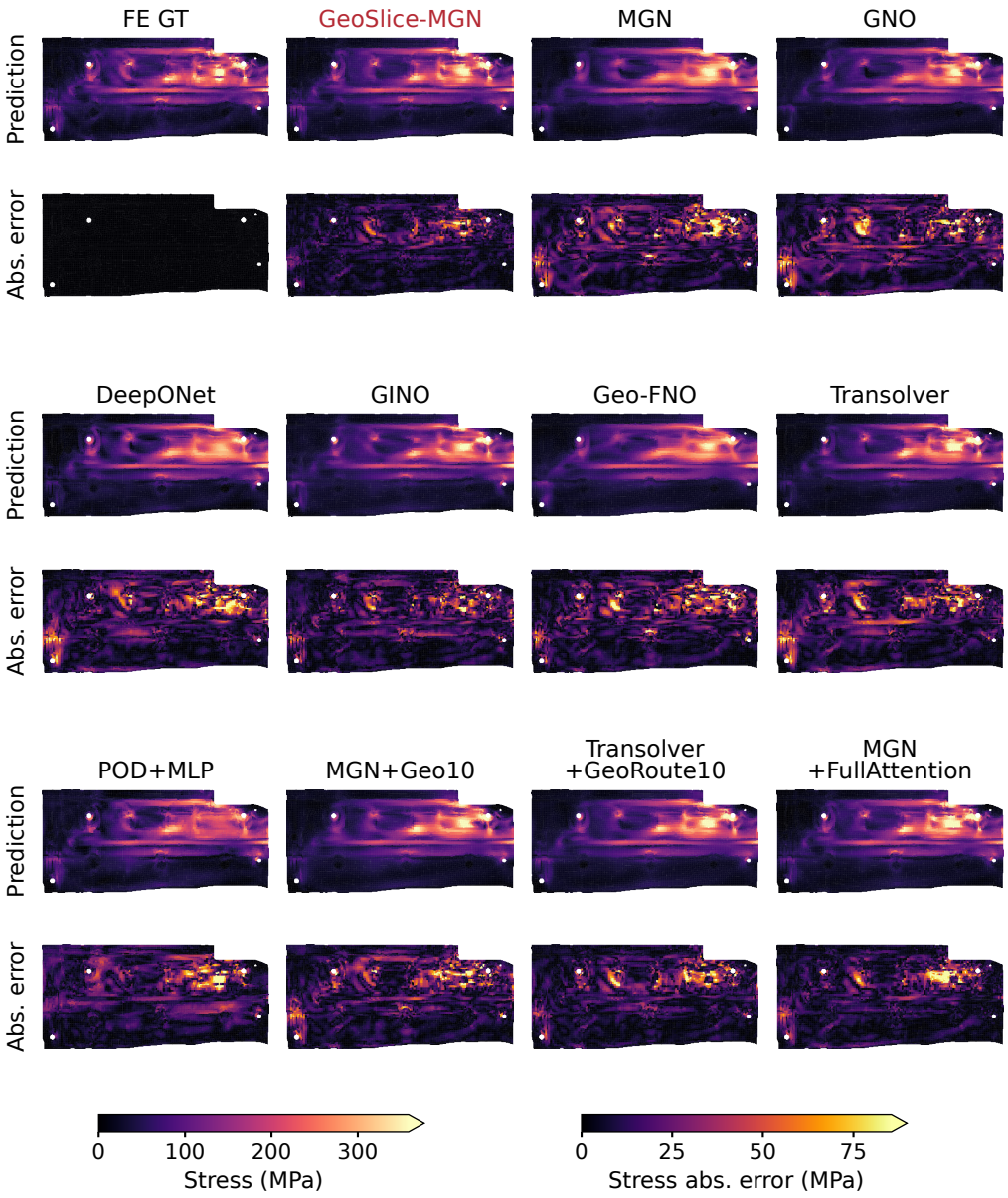


Figure 8: Effective stress $\sigma^{\star}_{\text{eff}}$ on `floorfrontdriver` and its absolute error $\left|\widehat{\sigma}^{\star}_{\text{eff}} - \sigma^{\star}_{\text{eff}}\right|$. FE GT has zero error by definition. The bottom-left color bar shows effective stress and the bottom-right bar shows its absolute error, both in MPa.

On `floorfrontR`, displacement is concentrated in the lower-left region (Fig. 9). GeoSlice-MGN recovers its position and extent. MGN, GNO, and Geo-FNO show larger deviations near the peak, while POD+MLP errors extend farther into the interior.

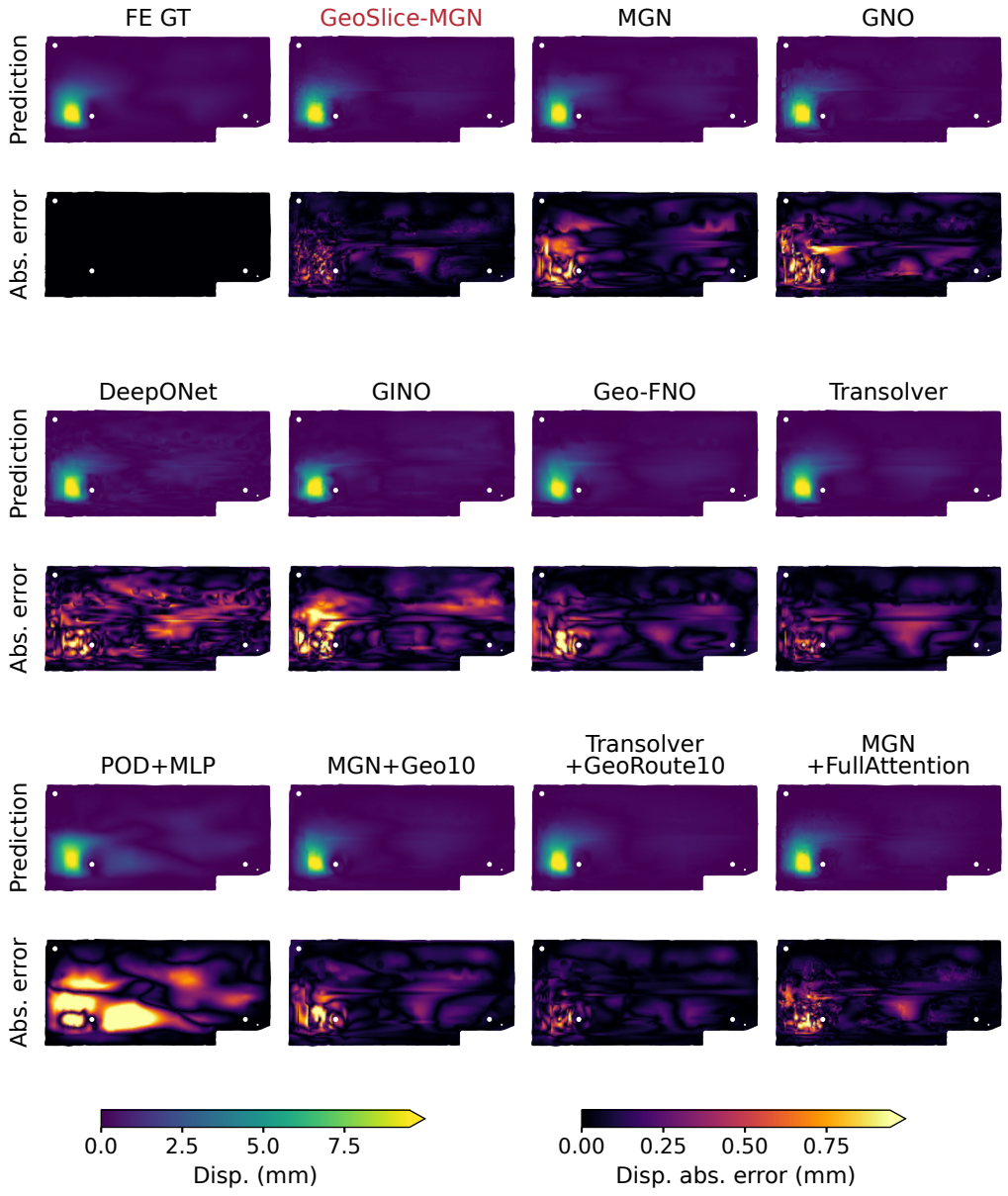


Figure 9: Nodal displacement magnitudes $\{\|\mathbf{u}_i^\star\|_2\}_{i=1}^{N_v}$ on `floorfrontR` and their node-wise absolute errors. The layout and color-bar conventions follow Fig. 7. Displacements and errors are measured in mm.

High stress on `floorfrontR` is also concentrated toward the lower left (Fig. 10). GeoSlice-MGN recovers that concentration with localized residual errors. MGN, GNO, and Geo-FNO have errors over a wider area.

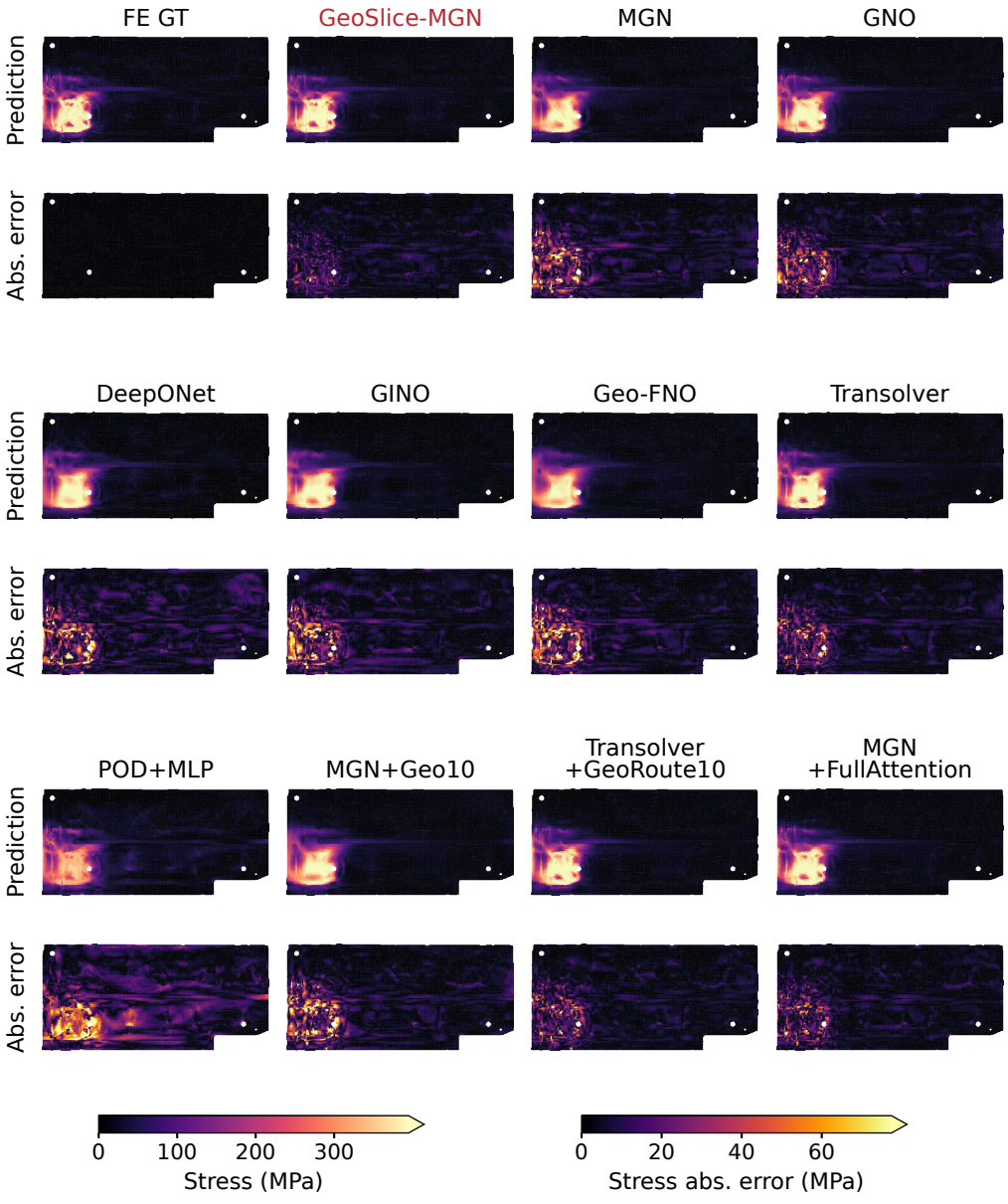


Figure 10: Effective stress $\sigma^{\star}_{\text{eff}}$ on `floorfrontR` and its absolute error. The layout and color-bar conventions follow Fig. 8. Stresses and errors are measured in MPa.

On `trunkfloor`, the high-displacement region extends from the lower left toward the left side and center (Fig. 11). MGN and GNO produce broad arc-shaped error patterns. Errors of GeoSlice-MGN cover a smaller area, with the largest deviations near the lower-left displacement peak.

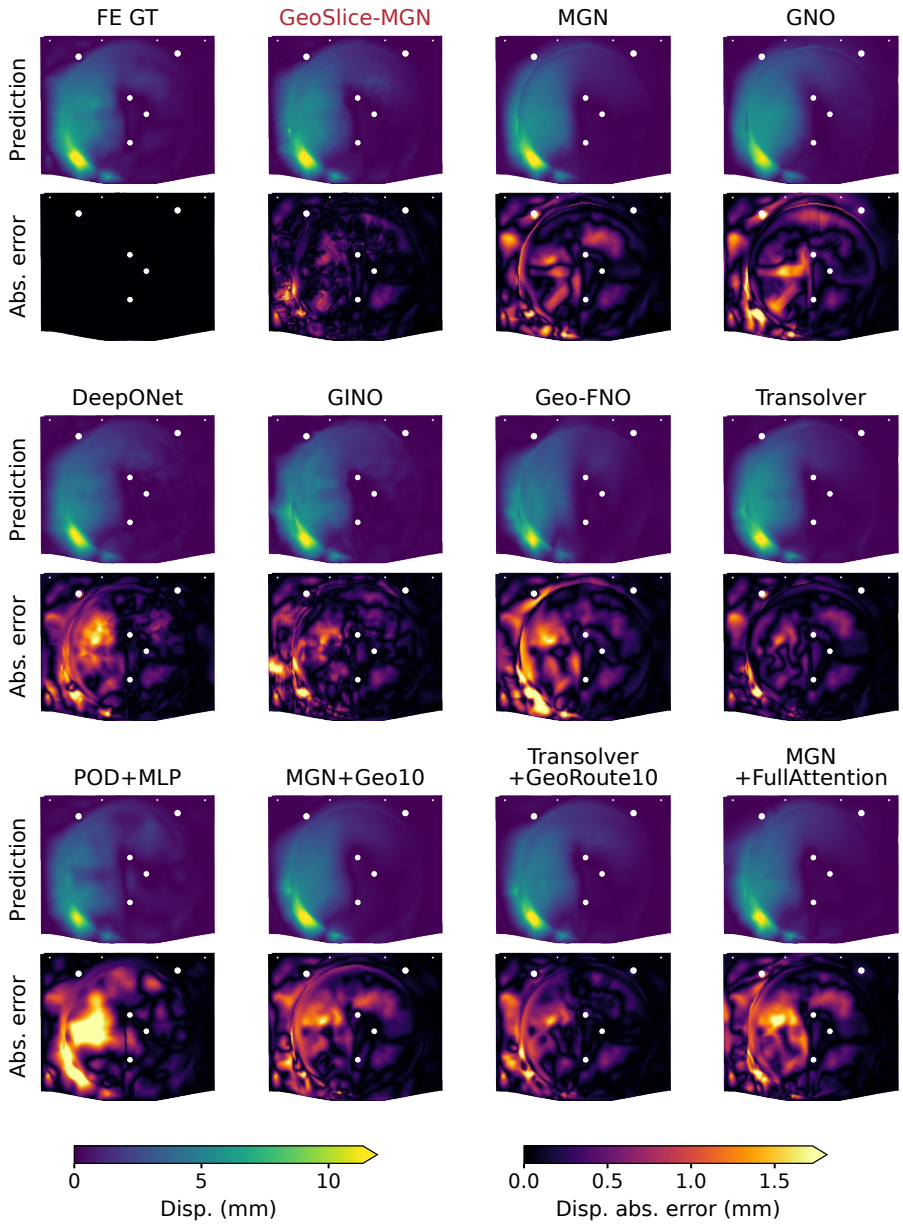


Figure 11: Nodal displacement magnitudes $\{\|\mathbf{u}_i^\star\|_2\}_{i=1}^{N_v}$ on `trunkfloor` and their node-wise absolute errors. The layout and color-bar conventions follow Fig. 7. Displacements and errors are measured in mm.

Stress on `trunkfloor` follows a left-side arc and localized interior patches (Fig. 12). Compared with the smoother predictions of MGN, GNO, and Transolver, GeoSlice-MGN retains more of this local variation. Errors remain around the high-stress regions in this case.

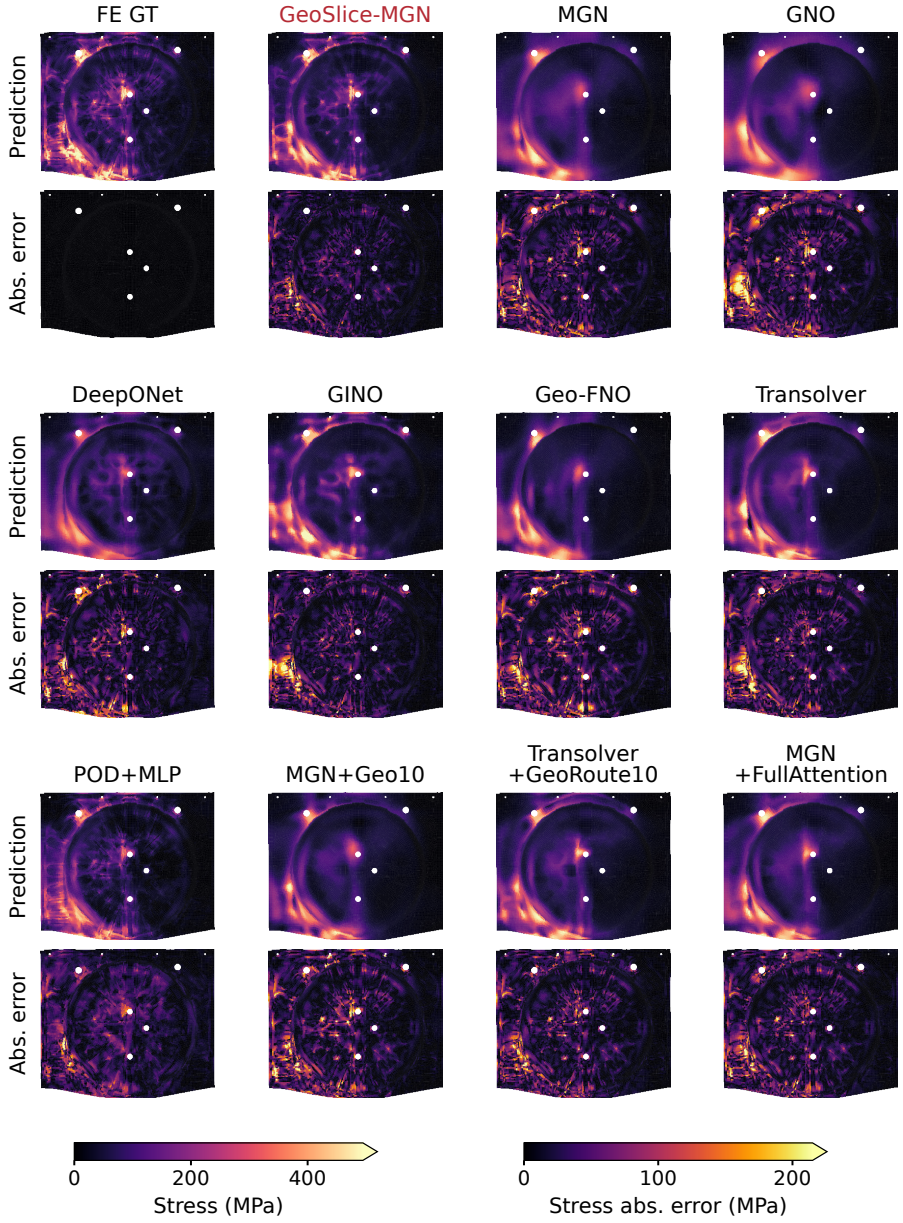


Figure 12: Effective stress $\sigma^{\star}_{\text{eff}}$ on `trunkfloor` and its absolute error. The layout and color-bar conventions follow Fig. 8. Stresses and errors are measured in MPa.

I Cross-Panel Adaptation

I.1 Transfer Task and Experimental Setup

A new or modified panel typically requires additional simulations and surrogate training. Reusing a model trained on an existing panel may reduce the amount of target data needed. We test one transfer direction, from `floorfrontR` to `floorfrontdriver`, comparing direct inference on the target mesh with adaptation using limited target cases. The panels differ in mesh topology, holes, stiffening features, and local boundaries. Loading cases are sampled independently for each panel and are not paired across panels.

The panels lie on opposite sides of the vehicle centerline, so we first align their lateral orientation. For each of the three compared models, we reflect target coordinates, impact positions, velocities, and displacement vectors using $R = \mathrm{diag}(1, -1, 1)$:

$$\widetilde{\mathbf{x}}_i = R\mathbf{x}_i, \qquad \widetilde{\mathbf{p}} = R\mathbf{p}, \qquad \widetilde{\mathbf{v}} = R\mathbf{v}, \qquad \widetilde{\mathbf{u}}_i = R\mathbf{u}_i. \tag{55}$$

Relative geometry is recomputed after reflection. Material parameters, constraint indicators, and von Mises stress are unchanged, and predicted displacement is mapped back with $R^{-1} = R$. The transformation aligns coordinate orientation while preserving each panel's geometry and connectivity.

We compare three settings. Direct transfer freezes the source model and evaluates it on the target mesh. Training from scratch starts from random initialization and uses limited target data. Pretraining followed by fine-tuning initializes from the source model and updates all parameters using that same target data. Comparing training from scratch with fine-tuning isolates the benefit of source-model initialization at a fixed data budget.

Scratch training and fine-tuning both use 50 target training cases and 50 validation cases; every setting is evaluated on the same 50 test cases. Each setting retains the corresponding source model's input statistics and output scales. Errors are computed in physical units after denormalization. Target checkpoints minimize validation normalized MSE, and all models use the same seed.

I.2 Cross-Panel Prediction and Adaptation Results

Table 12 compares test displacement and stress errors for MGN, Transolver, and GeoSlice-MGN under the three settings.

Table 12: Test relative L_2 errors for `floorfrontR`→`floorfrontdriver`. Direct transfer freezes the source model; scratch training and fine-tuning share the target training cases. Results use one common seed. Bold indicates the lowest error in each column.

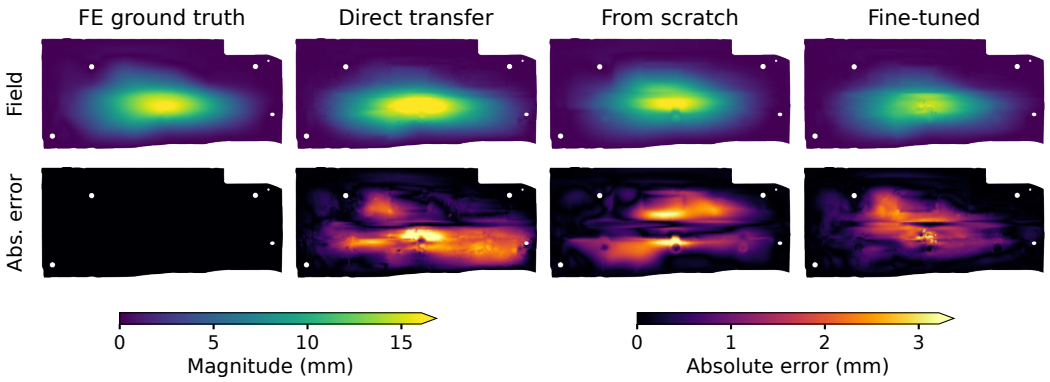
	Direct transfer		Target training from scratch		Pretraining + fine-tuning	
Model	Displacement	Stress	Displacement	Stress	Displacement	Stress
MGN	0.4828	0.3638	**0.3869**	**0.3915**	0.2192	0.2619
Transolver	0.4904	0.3491	0.4224	0.4308	0.1863	0.2357
GeoSlice-MGN	**0.4536**	**0.3288**	0.4139	0.4029	**0.1716**	**0.2333**

Direct transfer gives GeoSlice-MGN displacement and stress errors of 0.4536 and 0.3288. Full-parameter fine-tuning reduces them to 0.1716 and 0.2333, respectively. With the same amount of target training data, these errors are 58.5% and 42.1% below training from scratch. MGN and Transolver also benefit from pretraining. After adaptation, GeoSlice-MGN has the lowest errors of the three models, with a larger margin for displacement and a stress error close to Transolver's.

Updating only the GeoSlice-MGN decoders gives errors of 0.2091 and 0.2829, both higher than full-parameter fine-tuning. This comparison suggests that adaptation of intermediate representations contributes to the improvement in this transfer setting.

Figure 13 illustrates the same target case under the three settings. Fine-tuning recovers the main displacement concentration and stress patterns along stiffening features, reducing errors near holes and reinforcements relative to direct transfer and scratch training. These observations are consistent with the test-set comparison and support initializing from a source model when target data are limited. The evidence is specific to the reported transfer direction and seed.

(a) Nodal displacement magnitude (mm)



(b) von Mises effective stress (MPa)

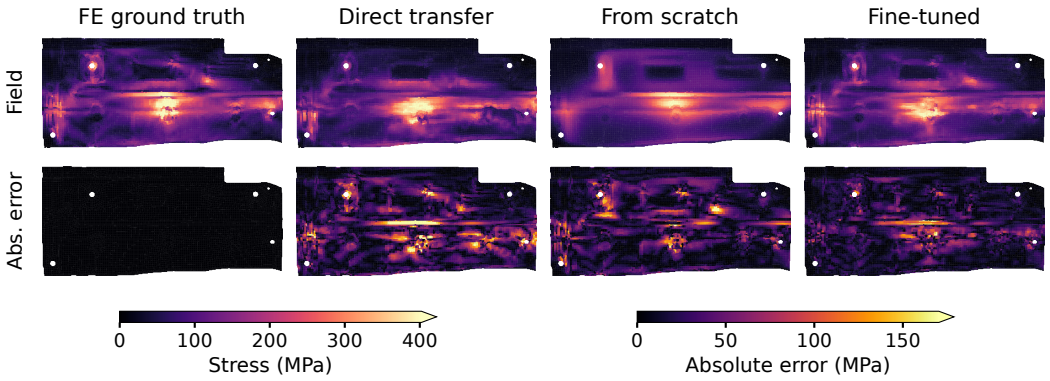


Figure 13: Predicted fields and errors of GeoSlice-MGN on the target panel. The first column shows FE ground truth; the remaining columns show predictions under the three settings in Table 12: direct transfer, target-panel training from scratch, and pretraining followed by fine-tuning. All settings use the same coordinate convention; training from scratch and fine-tuning use the same target training cases. Within each group, the top row shows ground truth or predictions, and the bottom row shows node-wise absolute displacement-magnitude errors or element-wise absolute effective-stress errors. All settings use the same field and error color scales for each output.

J COMPUTATIONAL EFFICIENCY

We compare inference time and peak GPU memory for all 11 configurations in Table 1 on the same device and under a common protocol. We also compare the GeoSlice-MGN forward-pass time with the FE simulation time for each panel.

J.1 SURROGATE INFERENCE COST

We measure each configuration in a separate process on an NVIDIA Tesla V100S GPU. After 30 warm-up passes, we time 10 passes for each of the 50 test cases using synchronized CUDA events. Reported times are means and standard deviations over the resulting 500 passes.

Timing includes internal geometric feature computation, both output heads, and POD full-field reconstruction. Models and inputs are already on the GPU. We exclude data loading, host preprocessing, transfers, and output denormalization. Geometric caches retain their state after warm-up.

Peak memory is the maximum PyTorch `max_memory_allocated` over test cases. It includes model parameters, fixed buffers, the mesh, one case's inputs, and forward intermediates; POD bases count as buffers. Table 13 reports time and memory under these conditions.

Table 13: GPU forward-pass time and peak PyTorch allocated memory. Times are mean±standard deviation across repeated passes (ms/case); memory is in MiB. † denotes a control developed here.

	`floorfrontdriver`		`floorfrontR`		`trunkfloor`	
Model	GPU forward time	Memory	GPU forward time	Memory	GPU forward time	Memory
POD+MLP	0.35 ± 0.09	17.3	0.33 ± 0.09	22.2	0.35 ± 0.09	39.7
DeepONet	1.83 ± 0.15	41.3	2.53 ± 0.09	56.4	2.95 ± 0.10	63.4
GNO	17.91 ± 0.20	137.6	28.46 ± 0.29	212.4	33.95 ± 0.25	252.0
GINO	20.53 ± 0.13	633.9	27.78 ± 0.08	1005.6	31.73 ± 0.17	1204.6
Geo-FNO	4.24 ± 0.19	44.5	4.42 ± 0.15	60.4	4.64 ± 0.32	66.7
MGN	14.71 ± 0.23	187.0	23.22 ± 0.21	294.3	27.45 ± 0.24	349.7
MGN+Geo10†	15.04 ± 0.23	187.4	23.76 ± 0.25	294.8	27.88 ± 0.22	350.3
Transolver	10.03 ± 0.08	107.9	15.35 ± 0.11	163.7	18.32 ± 0.10	193.4
Transolver+GeoRoute10†	10.80 ± 0.12	115.4	16.38 ± 0.14	176.0	19.51 ± 0.15	207.8
MGN+FullAttention†	35.49 ± 0.36	132.6	84.72 ± 0.60	206.1	105.41 ± 0.70	244.3
GeoSlice-MGN	15.33 ± 0.37	138.6	22.28 ± 0.37	215.7	25.88 ± 0.21	265.8

GeoSlice-MGN requires 15.33–25.88 ms per GPU forward pass and 138.6–265.8 MiB of peak memory. It is 2.32–4.07 times faster than FullAttention under this protocol. Together with Table 1, these measurements show the tradeoff between accuracy and computational cost: GeoSlice-MGN has lower displacement and stress errors than POD+MLP, DeepONet, and Geo-FNO, but requires longer inference times.

J.2 TIME COMPARISON WITH FINITE ELEMENT SIMULATION

Let $t_{\mathrm{FE}}^{(g)}$ denote the mean LS-DYNA simulation time for panel g, and $t_{\mathrm{NN}}^{(g)}$ the mean GPU forward-pass time of GeoSlice-MGN. Define

$$\mathrm{Speedup}^{(g)} = \frac{t_{\mathrm{FE}}^{(g)}}{t_{\mathrm{NN}}^{(g)}}. \tag{56}$$

The numerator measures a complete FE simulation, while the denominator measures one surrogate forward pass for peak displacement and stress. The ratio therefore compares the time needed to obtain the target response under each method's execution conditions. It excludes the costs of generating training data and training the surrogate.

Table 14: Per-case runtime comparison between GeoSlice-MGN and LS-DYNA. Network time covers a GPU forward pass; FE time covers a complete simulation.

Panel	Nodes	Shell elements	LS-DYNA (s/case)	GeoSlice-MGN (ms/case) / speedup
`floorfrontdriver`	7,408	7,374	50.73	15.33 ± 0.37 / $3,310\times$
`floorfrontR`	12,011	12,055	178.25	22.28 ± 0.37 / $8,000\times$
`trunkfloor`	14,440	14,589	80.25	25.88 ± 0.21 / $3,101\times$