IMPACTFORMER: A SPACE–TIME FACTORIZED NEURAL OPERATOR FOR HETEROGENEOUS FINITE-ELEMENT SIMULATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning a single surrogate across finite-element simulation datasets is challenging because their geometries, meshes, design parameterizations, and time grids differ. We study this problem for vehicle crash, one of the most demanding structural simulations, and introduce ImpactFormer, a neural operator trained jointly across crash datasets to map undeformed geometry and design parameters to complete spatiotemporal displacement fields. Each simulation is encoded by anchor tokens sampled from its structural mesh and a variable-length set of named parameter tokens, processed by a shared attention backbone with dataset-specific feature-wise modulation, so that datasets with different meshes and parameterizations share one conditional model. A space–time factorized head predicts simulation- and node-specific spatial coefficients of a single learned temporal basis shared across nodes, simulations, and datasets, so that one temporal representation serves datasets with different durations and frame counts and reconstructs complete trajectories in a single forward pass. We train a 19.2M-parameter model on 4,641 full-vehicle frontal crash simulations from five datasets: the CarCrashNet Neon, Yaris, and Silverado design-of-experiments tracks, the geometry-morphing SHIFT-Crash dataset, and Yaris-yield, a yield-strength dataset generated for this work. These datasets span approximately 350k to over 1M nodes, 12 to 40 predicted time frames, and three to six design parameters. Under a fixed evaluation protocol, ImpactFormer outperforms all state-of-the-art neural solvers on Neon, Yaris, Silverado and SHIFT-Crash: it reduces RMSE by up to 12.1% relative to the strongest of them, CrashSolver, and by 23% to 49% on average relative to AB-UPT, FIGConvUNet, Transolver, GeoTransolver and GAOT. It remains the most accurate neural solver on the complete meshes of up to 1.07M nodes, where it is up to 15% more accurate than CrashSolver retrained at full resolution. Pretrained only on sedans, ImpactFormer also adapts to a completely unseen pickup truck from a handful of simulations. Sixteen simulations of the truck reduce its error by 88%, and with four simulations its error is less than half that of the same network trained from scratch. Combined with a classical reduced-order model, ImpactFormer reduces error by up to 29% relative to CrashSolver on fixed meshes.

1 INTRODUCTION

Finite-element simulation is central to structural design, and vehicle crashworthiness is one of its most demanding applications: engineers must repeatedly evaluate how structural changes affect deformation during impact. Explicit finite-element simulations play a central role in this process, allowing engineers to study complex interactions involving contact, plasticity, and material failure (Belytschko et al., 1984). However, these simulations are computationally expensive. Full-vehicle models contain 10^5 to 10^6 nodes, with a single full-frontal simulation requiring up to several hundred core-hours (Elrefaie et al., 2026). Repeating these simulations across a large design space limits the number of configurations that can be explored. This has motivated the development of learned surrogates that predict deformation fields in seconds, enabling faster evaluation of candidate designs (Kohar et al., 2021; Nabian et al., 2026; Akhare et al., 2026). The availability of crash simulation datasets provides new opportunities to develop and systematically evaluate these methods.

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.

CarCrashNet (Elrefaie et al., 2026) includes parametric design-of-experiments (DoE) studies across three vehicles with fixed geometries, while SHIFT-Crash (Luminary Cloud, 2026) provides over 5,000 simulations with geometry morphed between runs.

Despite recent progress, existing crash surrogates are often developed for a specific vehicle, mesh, and set of design parameters. A model trained in one setting therefore does not readily transfer to another. Fixed-length parameter vectors (Jing et al., 2026) further complicate joint training when datasets differ in the number or physical meaning of their parameters. Differences in simulation duration and output frame counts introduce additional challenges for learning a shared temporal representation. Moreover, methods that predict deformation through autoregressive time stepping can accumulate errors over the crash trajectory. These limitations make it difficult to reuse simulation data across studies and develop models with broader applicability. In contrast, large-scale datasets and pretraining have driven substantial progress in computer vision and language (Deng et al., 2009; Bommasani et al., 2021), motivating related efforts in scientific machine learning (Subramanian et al., 2023; Herde et al., 2024; Elrefaie et al., 2024).

Beyond generalization, learned surrogates must also be evaluated against established approaches to reducing simulation costs. Classical reduced-order models remain effective for fixed-mesh parametric studies in structural design (Berkooz et al., 1993; Quarteroni et al., 2016; Benner et al., 2015). Comparing neural operators with these methods is therefore important for understanding where a more flexible model provides a practical advantage and where a simpler surrogate remains sufficient. These considerations motivate a central question:

Can a single model learn to accurately predict complete crash trajectories across vehicle types, mesh resolutions, design parameterizations, and temporal discretizations, rather than requiring a separate surrogate for each dataset?

To address this question, we introduce **ImpactFormer**, a 19.2M-parameter neural operator that learns from five crash datasets to predict complete spatiotemporal displacement fields (Figure 1). ImpactFormer combines heterogeneous input conditioning with a shared learned temporal representation to learn crash dynamics across vehicle types, mesh resolutions, design parameterizations, and temporal discretizations. Geometry tokens, variable-length named parameter tokens, and dataset-specific modulation provide a unified interface for jointly training on these datasets. The decoder predicts node-specific coefficients of a learned temporal basis shared across nodes, simulations, and datasets, allowing trajectories with different durations and output frame counts to use the same temporal representation. Together, these components extend vehicle-specific prediction to a single model that learns from heterogeneous crash data using one set of weights.

As in anchor–query architectures such as AB-UPT and CViT (Alkin et al., 2025; Wang et al., 2025b), a query cross-attention decoder evaluates the output at any mesh node. What is specific to ImpactFormer is sharing across heterogeneous datasets: named parameter tokens and dataset-conditioned modulation give different parameterizations one input interface, and the decoder produces run- and node-specific coefficients of one temporal basis shared by all datasets across their different durations and frame counts (Section 4).

Our main contributions are summarized as follows:

- **Joint learning across vehicles, meshes, and design parameterizations.** We train a single model on 4,641 simulations from five datasets, covering three vehicles, fixed and morphing meshes, and three to six design parameters. ImpactFormer achieves the lowest test RMSE of all state-of-the-art neural solvers on Neon, Yaris, Silverado and SHIFT-Crash, up to 12.1% below CrashSolver and 23–49% below AB-UPT, FIGConvUNet, Transolver, GeoTransolver and GAOT on average; on complete meshes of up to 1.07M nodes it is 11–15% more accurate than CrashSolver retrained at full resolution on the three vehicles (Section 5.1).
- **A temporal basis shared across heterogeneous datasets.** One learned temporal basis serves all five datasets across their different durations and frame counts. Controls on the same backbone (Section 5.3) show that it reduces error by up to 41% relative to both autoregressive rollout and a direct multi-frame head, and matches a decoder conditioned on time at one decoder pass per node instead of one per frame.
- **A systematic comparison with classical reduced-order modeling.** CrashDoE, a POD-plus-regression baseline fitted in CPU-minutes, is the most accurate model on fixed meshes

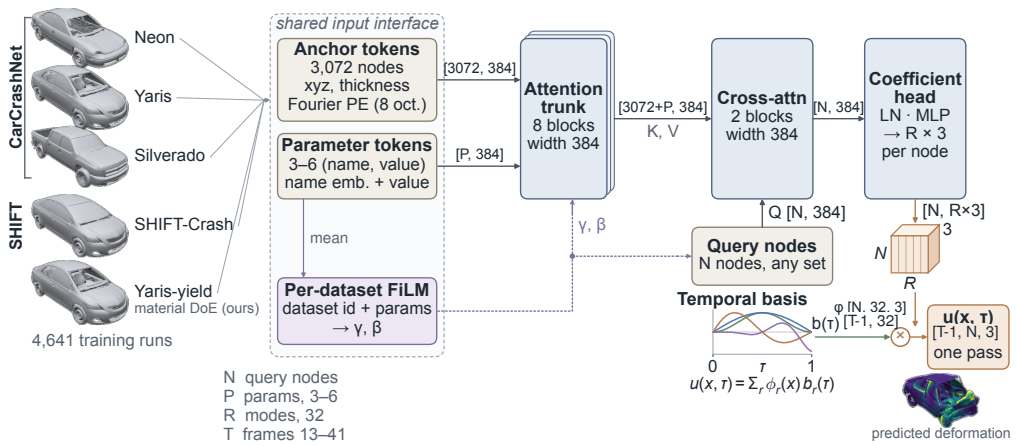

Figure 1: The ImpactFormer architecture. Anchor tokens sampled from the undeformed geometry and variable-length parameter tokens pass through an attention trunk with per-dataset FiLM modulation; each query node cross-attends to the trunk and emits $R\times3$ coefficients on a shared temporal basis $b(\tau)$ of R functions of the normalized time τ.

with few training runs, but ImpactFormer outperforms it on the larger Yaris dataset and on the geometry-morphing SHIFT-Crash dataset (Section 5.2). The two are complementary: combined, they reduce error by 10–29% relative to CrashSolver on the three fixed-mesh vehicles (Section 5.4).

- **Few-shot adaptation to an unseen vehicle.** To test how well ImpactFormer generalizes to a new vehicle type, we pretrain it only on sedans and evaluate it on a completely unseen pickup truck, the Silverado. Fine-tuning on 16 simulations reduces the error by 88% compared with zero-shot prediction, and fine-tuning on 4 simulations gives less than half the error of the same network trained from scratch on those simulations (Section 5.5).

2 Related Work

Surrogate models for crash simulation. Surrogate models have been used to predict crash response and deformation (Kohar et al., 2021), guide crash-box design optimization (Borse et al., 2023), and estimate collision damage (Liu et al., 2025). Mesh-based approaches include recurrent graph models for vehicle panels (Li et al., 2026b) and the body-in-white study of Nabian et al. (2026), which evaluates MeshGraphNet (Pfaff et al., 2021) and Transolver (Wu et al., 2024) using both time-conditioned prediction and autoregressive rollout. Akhare et al. (2026) apply GeoTransolver to bumper-beam and body-in-white crash datasets and compare one-shot, time-conditional and autoregressive temporal recipes; time-conditional prediction is most accurate on the bumper beam, and one-shot is competitive on both at much lower per-epoch training cost. Geometric generalization has also been investigated at the component level. Mask-Morph Graph U-Net combines coarse-graph morphing with masked pretraining and fine-tuning to predict terminal displacement fields across varying component geometries (Li et al., 2026a). A separate line of work retains the finite-element solver and replaces selected components or subdomains with learned models, including neural joint models, Finite Element Method Integrated Networks, and local/global coupling for spot-welded structures (André et al., 2023; Thel et al., 2024; 2025; Pulikkathodi et al., 2025).

Learning structural deformation and mechanics. Learned models of solid mechanics also address architectural structures (Pastrana et al., 2025), cloth (Wandel et al., 2025), deformation and contact (Feng et al., 2026; Wang et al., 2025a), hyperelastic materials (Liu et al., 2026), volumetric properties (Dagli et al., 2026), fluid–solid interaction (Tao et al., 2026), fracture (Wang et al., 2026), scale invariance (Lee et al., 2026) and adaptive precision (van den Dool et al., 2025).

Neural operators and learning across datasets. Graph-network simulators learn dynamics on particle and mesh discretizations (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2021; Hoang et al.,

2026). Neural operators formulate prediction as a mapping between input and output functions, as in DeepONet and the Fourier Neural Operator (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023). Architectures developed for geometry-dependent field prediction include Transolver and Transolver++ (Wu et al., 2024; Luo et al., 2025), GeoTransolver (Adams et al., 2025), AB-UPT (Alkin et al., 2025), GAOT (Wen et al., 2025), and adaptive Mamba neural operators (Song & Jiang, 2026). FIGConvUNet takes a convolutional approach using factorized implicit grids and global convolutions (Choy et al., 2025). Attention over heterogeneous inputs and latent or anchor tokens also underlies GNOT, Perceiver IO and UPT (Hao et al., 2023; Jaegle et al., 2022; Alkin et al., 2024), and CViT uses query-wise cross-attention to evaluate continuous output fields (Wang et al., 2025b), while Schwarz Neural Inference combines local neural operators through domain decomposition (Huang et al., 2026). Continuous-time and multi-step operators avoid the error accumulation of autoregressive rollout (Hou et al., 2026; Thompson et al., 2026).

Training on multiple physical systems has been explored in scientific foundation models (Subramanian et al., 2023; McCabe et al., 2024; Hao et al., 2024; Zhou et al., 2025; Herde et al., 2024; Holzschuh et al., 2026). For mesh-based simulation, MeshMask demonstrates pretraining across multiple CFD datasets (Garnier et al., 2025), and Zhang et al. (2026) learn geometric representations from unlabeled shapes before training on physical fields. Benchmarks such as PDEBench, The Well, and RealPDEBench support evaluation across a range of physical systems (Takamoto et al., 2022; Ohana et al., 2024; Hu et al., 2026). In automotive applications, DrivAerNet++ and CarBench support aerodynamic prediction (Elrefaie et al., 2024; 2025), while CarCrashNet and SHIFT-Crash provide crash simulation data (Elrefaie et al., 2026; Luminary Cloud, 2026). Our focus is joint learning from crash datasets with different vehicles, meshes, and parameterizations, using one model and a shared temporal basis to predict complete displacement trajectories across all five datasets.

Reduced-order models and comparisons with learned surrogates. Proper orthogonal decomposition provides an established way to represent simulation fields in a low-dimensional basis (Berkooz et al., 1993; Benner et al., 2015; Quarteroni et al., 2016; Le Guennec et al., 2018). Learned and classical reduction are not disjoint approaches: Friedl et al. (2025) learn reduced Lagrangian dynamics on nonlinear latent manifolds, OrthoSolver learns nonlinear modes with orthogonality regularization (Pang et al., 2026), and subspace regression predicts parameter-dependent reduced bases (Fanaskov et al., 2026). Direct comparisons also exist for crash simulation. Defoort et al. (2025) compare a CUR-based reduced-order surrogate with neural fields and examine how their relative accuracy changes with the number of training simulations. Our evaluation considers both fixed and morphing meshes across multiple vehicles. We compare CrashDoE, six state-of-the-art neural solvers, and the jointly trained ImpactFormer to identify where each approach is effective (Sections 5.1 and 5.2).

3 The Datasets and the Benchmark

We train and evaluate on five crash-simulation datasets (see Figure 2): the Neon, Yaris and Silverado full-frontal tracks of CarCrashNet, SHIFT-Crash, and Yaris-yield, a material design of experiments generated for this work.

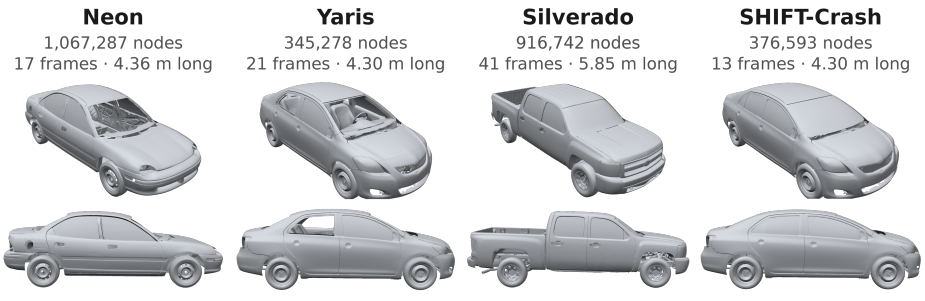

Figure 2: The four geometries of the five datasets, undeformed, each drawn at the same length (true lengths in the headers): three vehicles, of which the Yaris also underlies SHIFT-Crash. The CarCrashNet vehicles keep their mesh fixed across runs, whereas the SHIFT-Crash mesh is a morphed Yaris, reshaped per run.

Vehicles, geometry and design of experiments. We use the CarCrashNet dataset (Elrefaie et al., 2026), based on publicly available full-vehicle finite-element models of the 1996 Dodge Neon, the 2010 Toyota Yaris (coarse mesh), and the 2014 Chevrolet Silverado pickup (National Highway Traffic Safety Administration, 2026; Center for Collision Safety and Analysis, 2026). Within each category the mesh is identical across runs; what varies is the impact velocity v and two shell-thickness scale factors, s_{fs} for the front support and s_{lr} for the lower rail and subframe, over 50–64 km/h and 0.90–1.10. SHIFT-Crash (Luminary Cloud, 2026) starts from the same CCSA Yaris model and morphs it with a deformation cage: a single 376,593-node topology is reshaped for every run by four geometric parameters, and rail thickness and rail yield strength vary as well. Yaris-yield is a material DoE of our own (Appendix H): 119 runs placed by Latin hypercube sampling (McKay et al., 1979) over the three Yaris parameters and two yield-strength scale factors of $\pm 30\%$ on the front-support and lower-rail part groups. All runs are transient explicit solves with contact, plasticity and element failure (Belytschko et al., 1984), computed with OpenRadioss (OpenRadioss, 2026). SHIFT-Crash is used as released, with the split of Elrefaie et al. (2026).

Table 1: Statistics of the five datasets: 4,641 training runs. Scores use the pinned evaluation node set (Appendix G); T frames are exported and $T-1$ predicted. Parameters: (v, impact velocity; s_{fs}, s_{lr}, front-support and lower-rail thickness scales; t_{rail}, σ_{rail}, rail thickness and yield strength; σ_{fs}, σ_{lr}, yield-strength scales).

	Neon	Yaris	Silverado	SHIFT-Crash	Yaris-yield
train / val / test	49 / 25 / 25	353 / 46 / 44	56 / 4 / 15	4,088 / 518 / 504	95 / 12 / 12
Nodes (full structural)	1,067,287	345,278	916,742	376,593	345,278
Pinned eval nodes	296,171	210,621	254,422	229,722	345,278
Frames T ($T-1$)	17 (16)	21 (20)	41 (40)	13 (12)	21 (20)
t_{end} (s)	0.16	0.20	0.20	0.12	0.20
Parameters	3: v, s_{fs}, s_{lr}	3: v, s_{fs}, s_{lr}	3: v, s_{fs}, s_{lr}	6: four morphs, t_{rail}, σ_{rail}	5: v, s_{fs}, s_{lr}, σ_{fs}, σ_{lr}

Splits and pinned node sets. Splits are frozen per dataset and disjoint between training and test (Table 1). The metric is the RMSE of nodal displacement in mm over a fixed, pinned set of evaluation nodes per dataset and the $T-1$ predicted frames of a test run, averaged over the test runs.

Baselines. Four of the five datasets have neural baselines: the three CarCrashNet vehicles and SHIFT-Crash, which was released without neural-solver results; Elrefaie et al. (2026) report Transolver, GeoTransolver and CrashSolver results on it, which we include. On each of them we evaluate six state-of-the-art neural solvers, GAOT, Transolver, GeoTransolver, FIGConvUNet, AB-UPT and CrashSolver (Section 2), under the same evaluation protocol. CrashSolver, the strongest of them on all four datasets, is trained with three random seeds, and its best run serves as the reference for each dataset.

4 IMPACTFORMER

As shown in Figure 1, ImpactFormer maps the undeformed vehicle structure and design parameters to the complete spatiotemporal displacement field of the crash.

Anchor tokens. The anchor encoder samples 3,072 nodes of the undeformed structure, redrawn at random at every step (2,048 from the benchmark node set at evaluation), and embeds their normalized coordinates and a thickness channel (nodal shell thickness on SHIFT-Crash, a constant elsewhere) into attention tokens, so one input interface serves every mesh without a mesh graph. The anchors are the vehicle context of the trunk, not the output resolution: every output node is a query with its own coordinates, and the reported model distinguishes runs of a dataset through its parameter tokens rather than through the anchors.

Parameter tokens. Each design parameter becomes one (name, value) token, built from a learned embedding of the name and a shared encoding of the standardized value. Datasets with 3 to 6

parameters thus share one interface. Time enters only through $\tau = t/t_{\text{end}}$, so end times from 0.12 to 0.20 s share one temporal axis as well.

Attention trunk. Anchor and parameter tokens are concatenated and passed through a stack of self-attention blocks (Vaswani et al., 2017) of width 384 and depth 8. A per-dataset FiLM modulation (Perez et al., 2018) of the token embeddings conditions the shared trunk on the dataset; on a CarCrashNet vehicle, the only run-specific signal is three parameter tokens among 3,075.

Query cross-attention and the factorized space–time head. A query point x, which can be any node of the full mesh, is given a multi-octave positional encoding and cross-attends to the trunk output, and the decoder emits $R\times 3$ coefficients. The displacement field is then the factorized product

$$\hat{u}(x, \tau \mid G, p, d) \;=\; \sum_{r=1}^{R} \varphi_r(x \mid G, p, d)\, b_r(\tau), \qquad R = 32, \tag{1}$$

where $\hat{u} \in \mathbb{R}^3$ is the predicted nodal displacement of x at normalized time $\tau \in (0, 1]$ (the displacement at $\tau{=}0$ is zero by definition and is neither predicted nor scored) for a run with undeformed geometry G (represented by the anchors), parameter set p and dataset d; $\varphi_r \in \mathbb{R}^3$ is the r-th spatial coefficient produced by the decoder; and b_r is the r-th function of a temporal basis that is learned as a function of τ and shared by all nodes, runs and datasets. Unlike DeepONet, POD-DeepONet and separable operator networks, which combine a basis of the output coordinates with input-dependent coefficients (Lu et al., 2021; 2022; Yu et al., 2024; Mandl et al., 2025), the spatial coefficients here depend jointly on the query and on the run context, and a single temporal dictionary must serve five datasets with different durations and frame grids. For one run the predicted trajectory matrix has rank at most $\min(R, T{-}1)$, so R measures the size of the shared dictionary rather than the rank of any single trajectory. A single forward pass yields φ for every query node, and evaluating b on a dataset's own frame grid gives all $T{-}1$ output frames at once.

Training. The 19.2M-parameter ImpactFormer is trained jointly on the five datasets on the mean-squared error of the normalized nodal displacement at a random subset of query nodes per run. Datasets are sampled so that SHIFT-Crash, which holds 88% of the training runs, does not dominate; optimizer, schedule and sampling probabilities were selected on validation. One network trains in about 1.7 GPU-hours on an A100-80GB.

5 Results

5.1 Ranking against state-of-the-art neural solvers

ImpactFormer achieves the lowest test RMSE of all state-of-the-art neural solvers on Neon, Yaris, Silverado and SHIFT-Crash (Table 2). It is 12.1%, 9.0% and 8.0% more accurate than the strongest neural solver on Neon, Yaris and Silverado and 3.3% more accurate than the CrashSolver reference on SHIFT-Crash, and its margins over the other state-of-the-art architectures are far larger: on average 23% below AB-UPT, 33% below FIGConvUNet, 34% below Transolver, 37% below GeoTransolver and 49% below GAOT. The advantage is systematic: ImpactFormer is more accurate than the CrashSolver reference on 88%, 73%, 73% and 67% of the Neon, Yaris, Silverado and SHIFT-Crash test simulations, with $p < 10^{-3}$ on Neon, Yaris and SHIFT-Crash (Appendix G). Trained on SHIFT-Crash alone, ImpactFormer reaches 5.71 mm, 6.3% below the strongest neural solver (Appendix C.1).

Full-mesh evaluation. The benchmark scores a pinned subset (28 to 61% of the nodes), the resolution at which the neural solvers are trained. Because ImpactFormer decodes at arbitrary query nodes, it also predicts the complete meshes of up to 1.07M nodes, where it is the most accurate neural solver on every dataset (Table 2). Subset-trained solvers degrade 1.2- to 11.2-fold at full resolution, consistent with Sakarvadia et al. (2026), and on the three vehicles ImpactFormer is still 11–15% more accurate than CrashSolver retrained on the full mesh.

Table 2: Test RMSE of nodal displacement (mm, lower is better) on the pinned node set of the benchmark (Appendix G) and the full mesh of the same test runs. n: number of test simulations. Underlined: best baseline per column; bold: ImpactFormer, the most accurate neural solver in every column.

	Neon (n=25)		Yaris (n=44)		Silverado (n=15)		SHIFT-Crash (n=504)	
Model	pinned	full	pinned	full	pinned	full	pinned	full
nodes evaluated	*296k*	*1,067k*	*211k*	*345k*	*254k*	*917k*	*230k*	*377k*
GAOT	9.945	16.827	8.695	10.266	14.536	102.1	10.938	13.721
Transolver	6.482	65.125	16.089	54.553	8.792	98.577	6.749	35.288
GeoTransolver	4.973	21.514	6.266	22.121	44.250	59.112	7.828	47.979
FIGConvUNet	5.762	12.867	7.774	9.389	10.294	17.568	9.411	14.050
AB-UPT	4.073	16.301	8.628	11.443	10.451	47.151	6.938	10.920
CrashSolver	<u>3.959</u>	8.816	<u>6.026</u>	12.444	<u>8.171</u>	59.332	6.239	13.788
CrashSolver, full-mesh training	4.218	<u>7.071</u>	6.228	<u>6.326</u>	8.296	<u>8.986</u>	<u>6.088</u>	<u>6.903</u>
ImpactFormer	**3.481**	**6.308**	**5.481**	**5.561**	**7.521**	**7.662**	**6.031**	**6.892**

5.2 Two regimes: fixed versus morphing mesh

Where does a classical surrogate suffice? We evaluate a classical reduced-order model, CrashDoE, on both kinds of dataset (Table 3). It is a proper orthogonal decomposition of the training trajectories (Berkooz et al., 1993) with a regularized regression from the DoE parameters to the modal coefficients, fitted in CPU-minutes. With few training runs on a fixed mesh (Neon, 49; Silverado, 56) it is the most accurate model; with more runs (Yaris, 353) ImpactFormer overtakes it by 8%, and on SHIFT-Crash, where every run reshapes one shared node topology and the classical model can only be fitted on the six parameters, ImpactFormer is 10% more accurate.

Table 3: The two regimes (test RMSE, mm; training runs in parentheses; on SHIFT-Crash the classical entry is a probe on the shared topology). Bold: lowest entry of a row.

regime	dataset (runs)	classical ROM	CrashSolver	ImpactFormer
fixed mesh	Neon (49)	**2.845**	3.959	3.481
	Yaris (353)	5.982	6.026	**5.481**
	Silverado (56)	**7.322**	8.171	7.521
morphing mesh	SHIFT-Crash (4,088)	6.706	6.239	**6.031**

The classical model exists for one node topology and parameterization, whereas one ImpactFormer covers all five datasets; as the datasets also differ in vehicle and response complexity, these are observations rather than a causal boundary between the regimes.

5.3 Temporal-head controls

Four controls replace the shared temporal basis on the same trunk, everything else fixed (Table 4; three paired seeds unless stated). The shared basis is the most accurate one-pass decoder: a *direct multi-frame head* without a basis is up to 71% worse, and teacher-forced *autoregressive rollout* (Nabian et al., 2026; Akhare et al., 2026) is worse on every dataset and seed, by up to 70% on Silverado, the longest trajectory. A *time-conditioned decoder*, with τ in the query, is as accurate but needs $T-1$ decoder passes per node instead of one. Every dataset still improves up to R=32.

5.4 Combining ImpactFormer with a classical model

On a fixed mesh, the classical model and ImpactFormer are complementary. A convex combination of ImpactFormer and the classical field, with one weight per vehicle chosen on validation, is more accurate than either model on all three vehicles (Figure 3): 2.814, 5.404 and 6.420 mm, 1.1, 1.4

Table 4: Change in test RMSE when the shared temporal basis (R=32) is replaced, same trunk; mean of paired per-seed changes.

	Neon	Yaris	Silverado	SHIFT-Crash	Yaris-yield
time-conditioned decoder ($T-1$ passes)	+1%	−1%	+5%	+1%	−4%
direct multi-frame head	+71%	+4%	+23%	+5%	+6%
autoregressive rollout	+33%	+34%	+70%	+16%	+10%
shared basis of size 4 instead of 32 (one seed)	+31%	+39%	+55%	+13%	+13%

and 12.3% below the better component and 10 to 29% below the CrashSolver reference. Training the operator on the classical residual instead gains almost nothing (Appendix F.4).

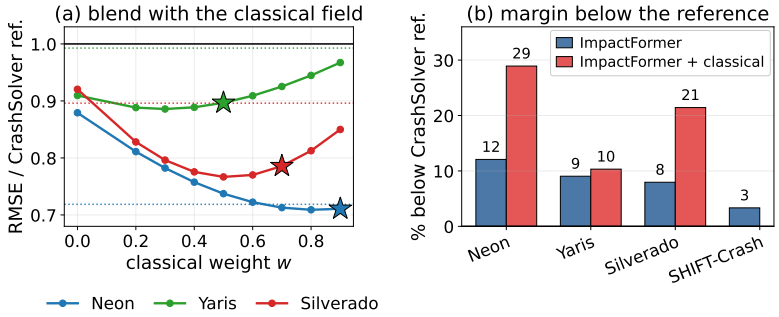

Figure 3: ImpactFormer combined with the classical field: (a) blend with weight w on the classical field, relative to the CrashSolver reference (stars: weight selected on validation; dotted: classical model alone); (b) margins below the CrashSolver reference, which on SHIFT-Crash was trained on a different node subset.

5.5 Adaptation and qualitative results

A new vehicle from few simulations. To test out-of-distribution generalization to a new vehicle type, we pretrain ImpactFormer only on sedans and withhold the Silverado pickup truck completely. Zero-shot prediction of this unseen vehicle type fails (135.2 mm). Fine-tuning the pretrained ImpactFormer changes this, as shown in Table 5: four simulations cut the error fivefold and sixteen by 88%, to 16.2 mm, after about 11 minutes of fine-tuning on one GPU. Pretraining is what makes this possible: with four simulations, the fine-tuned model has less than half the error of the same network trained from scratch (more analysis is given in Appendix C.2).

Table 5: **ImpactFormer adapts to an unseen vehicle type: the Chevrolet Silverado pickup.** Pretrained on sedans, the model is adapted using 4, 8, or 16 Silverado simulations, with fine-tuning taking approximately 11 minutes on a single GPU. Test RMSE (mm) on 15 held-out Silverado simulations, three seeds (mean ± s.d.; zero-shot: mean and range).

Silverado runs used	0 (zero-shot)	4	8	16	56 (joint)
test RMSE (mm)	135.2 (123.5–143.6)	26.4 ± 0.5	20.1 ± 0.3	**16.2 ± 0.7**	7.52
reduction vs. zero-shot	–	80%	85%	**88%**	94%

Qualitative results. Figure 4 shows one held-out test run for each dataset with neural baselines. ImpactFormer accurately reproduces the crash in all four datasets, from the global motion of the body to the folded front structure, and keeps the cabin and rear body within a few millimeters of the finite-element solution; the remaining error is confined to the most strongly folded parts of the crush zone and the front wheels.

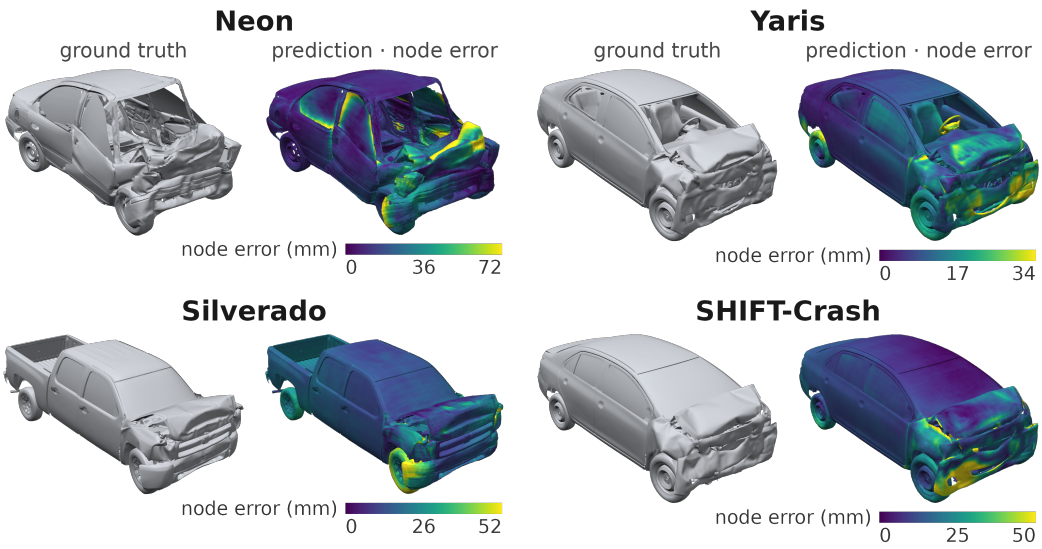

Figure 4: One held-out test run of Neon, Yaris, Silverado and SHIFT-Crash at the final frame: ground truth (left) and the ImpactFormer prediction colored by node error (right). The predicted deformation closely matches the finite-element result, including the crushed front structure.

6 Conclusion and Limitations

In this paper, we introduced ImpactFormer, a neural operator for data-driven crash simulation across different vehicles, meshes, and design parameterizations. Trained on 4,641 simulations from five datasets, the 19.2M-parameter model covers three vehicles, on fixed and morphed meshes, using a single set of weights. By combining mesh-based geometry tokens and named design parameters with a factorized space–time decoder, ImpactFormer predicts complete displacement trajectories at arbitrary mesh nodes in a single forward pass, without autoregressive time stepping.

In our benchmark, ImpactFormer is the most accurate neural solver on Neon, Yaris, Silverado and SHIFT-Crash, reducing test RMSE by up to 12.1% relative to the strongest state-of-the-art neural solver and by 23–49% on average, and up to 83%, relative to the others. Controls on the same backbone show that the shared temporal basis reduces error by up to 41% relative to autoregressive rollout at a single decoder pass per node, and combining ImpactFormer with a classical reduced-order model reduces error by up to 29% relative to CrashSolver on fixed meshes.

ImpactFormer supports rapid design sweeps, full-field prediction on meshes of more than one million nodes, and adaptation to an unseen vehicle type. Pretrained only on sedans and fine-tuned with sixteen simulations of a pickup truck, it reduces the RMSE on the truck by 88%, from 135.2 to 16.2 mm, and with four simulations, pretraining more than halves the error of training the same network from scratch.

Several challenges remain. Accurate zero-shot prediction of unseen vehicles is not yet achieved, and the three vehicles considered here cover only a limited range of structural designs and one load case, full-frontal impact. The yield-strength experiments also show that the direction of material-dependent changes is captured more reliably than their magnitude and shape. Although this work focuses on full-frontal vehicle crash, the approach is not specific to vehicles: anchor tokens on arbitrary meshes, named parameter tokens, and a shared temporal basis apply equally to other finite-element and physics simulations whose datasets differ in mesh size, number of time frames, material properties, and boundary conditions. We believe ImpactFormer provides a foundation for more accurate and reusable structural surrogates, supporting faster design exploration and reducing reliance on repeated high-cost crash simulations.

AI USE STATEMENT

We used generative AI tools to assist with manuscript proofreading, to improve grammar, clarity, and readability, and to help with reference formatting and title suggestions. All simulation data were generated with OpenRadioss, not generative AI, and no AI tools were used for translation. The authors reviewed all AI-assisted work, checked the code and references, verified reported results against experiment logs, and take full responsibility for the final content.

ETHICS STATEMENT

This work involves no human subjects and no personal data. All data are finite-element crash simulations of publicly available vehicle models. The surrogate is intended for design-stage exploration; it does not replace validated crash simulation or physical testing in any safety assessment, and its errors on designs and vehicles outside the training range (Section 5.5, Appendices C, E and G) must be taken into account by anyone who uses it.

REPRODUCIBILITY STATEMENT

Every benchmark number for our models and the reference baseline follows one fixed evaluation protocol: pinned node subsamples on which both are scored, frozen splits, the reference neural solver run at three seeds, and every benchmark selection of our models made on validation. Appendix A describes the model configuration, and Appendix G the metric, node sets and baseline references that define the benchmark.

REFERENCES

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

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. *arXiv preprint arXiv:2605.27758*, 2026.

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal Physics Transformers: A framework for efficiently scaling neural operators. In *Advances in Neural Information Processing Systems*, volume 37, pp. 25152–25194, 2024.

Benedikt Alkin, Maurits Bleeker, Richard Kurle, Tobias Kronlachner, Reinhard Sonnleitner, Matthias Dorfer, and Johannes Brandstetter. AB-UPT: Scaling neural CFD surrogates for high-fidelity automotive aerodynamics simulations via anchored-branched universal physics transformers. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=nwQ8nitlTZ.

Victor André, Miguel Costas, Magnus Langseth, and David Morin. Neural network modelling of mechanical joints for the application in large-scale crash analyses. *International Journal of Impact Engineering*, 177:104490, 2023.

Ted Belytschko, Jerry I. Lin, and C. S. Tsay. Explicit algorithms for the nonlinear dynamics of shells. *Computer Methods in Applied Mechanics and Engineering*, 42(2):225–251, 1984.

Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

Gal Berkooz, Philip Holmes, and John L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:539–575, 1993.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Aditya Borse, Rutwik Gulakala, and Marcus Stoffel. Machine learning enhanced optimisation of crash box design for crashworthiness analysis. *PAMM*, 23(4):e202300145, 2023.

Center for Collision Safety and Analysis. Finite element models. Official model repository, 2026. URL https://www.ccsa.gmu.edu/models/. Accessed: 2026-05-04.

Chris Choy, Alexey Kamenev, Jean Kossaifi, Max Rietmann, Jan Kautz, and Kamyar Azizzadenesheli. Factorized implicit global convolution for automotive computational fluid dynamics prediction. *arXiv preprint arXiv:2502.04317*, 2025.

Rishit Dagli, Donglai Xiang, Vismay Modi, Charles Loop, Clement Fuji Tsang, Anka He Chen, Anita Hu, Gavriel State, David I. W. Levin, and Maria Shugrina. VoMP: Predicting volumetric mechanical property fields. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4ea4a1ea4d9ff273688c8e92bd087112-Abstract-Conference.html.

Thibaut Defoort, Yves Le Guennec, Jose Vicente Aguado, and Domenico Borzacchiello. Comparing traditional surrogate modelling and neural fields for vehicle crash simulation data. In *SIA Simulation numérique 2025*, Guyancourt, France, 2025. URL https://hal.science/hal-05097364v1.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Mohamed Elrefaie, Florin Morar, Angela Dai, and Faez Ahmed. DrivAerNet++: A large-scale multimodal car dataset with computational fluid dynamics simulations and deep learning benchmarks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 499–536, 2024.

Mohamed Elrefaie, Dule Shu, Matt Klenk, and Faez Ahmed. CarBench: A comprehensive benchmark for neural surrogates on high-fidelity 3D car aerodynamics. *arXiv preprint arXiv:2512.07847*, 2025.

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.

Vladimir Fanaskov, Vladislav Trifonov, Alexander Rudikov, Ekaterina Muravleva, and Ivan Oseledets. Deep learning for subspace regression. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/bb568477d056a8a9293c0ba0bbb8354f-Abstract-Conference.html.

Zhe Feng, Shilong Tao, Haonan Sun, Shaohan Chen, Zhanxing Zhu, and Yunhuai Liu. MAVEN: A mesh-aware volumetric encoding network for simulating 3D flexible deformation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6920c76183516fb8efff75416393a577-Abstract-Conference.html.

Katharina Friedl, Noémie Jaquier, Jens Lundell, Tamim Asfour, and Danica Kragic. A Riemannian framework for learning reduced-order Lagrangian dynamics. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/913c7cbda1811776ff3db7316fd86808-Abstract-Conference.html.

Paul Garnier, Vincent Lannelongue, Jonathan Viquerat, and Elie Hachem. MeshMask: Physics-based simulations with masked graph neural networks. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9e443458a6691654f0b163915ce14979-Abstract-Conference.html.

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.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17616–17635. PMLR, 2024.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 72525–72624, 2024.

Tai Hoang, Alessandro Trenta, Alessio Gravina, Niklas Freymuth, Philipp Becker, Davide Bacciu, and Gerhard Neumann. Improving long-range interactions in graph neural simulators via Hamiltonian dynamics. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5b6346a05a537d4cdb2f50323452a9fe-Abstract-Conference.html.

Benjamin Holzschuh, Georg Kohl, Florian Redinger, and Nils Thuerey. P3D: Highly scalable 3D neural surrogates for physics simulations with global context. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/bf67feee86c1711d63cb4e6dc9998702-Abstract-Conference.html.

Xianglong Hou, Xinquan Huang, and Paris Perdikaris. CFO: Learning continuous-time PDE dynamics via flow-matched neural operators. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8bfbf4ec87e1e331f0b1adc483b53b6b-Abstract-Conference.html.

Peiyan Hu, Haodong Feng, Hongyuan Liu, Tongtong Yan, Wenhao Deng, Tianrun Gao, Rong Zheng, Haoren Zheng, Chenglei Yu, Chuanrui Wang, Kaiwen Li, Zhi-Ming Ma, Dezhi Zhou, Xingcai Lu, Dixia Fan, and Tailin Wu. RealPDEBench: A benchmark for complex physical systems with real-world data. In *International Conference on Learning Representations*, 2026.

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

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=fILj7WpI-g.

Cheng Jing, Uvini Balasuriya Mudiyanselage, Abhishek Verma, Kallol Bera, Shahid Rauf, and Kookjin Lee. Extending Fourier neural operators for modeling parameterized and coupled PDEs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c253f6fe5cc2bd1905427da63ac37917-Abstract-Conference.html.

Christopher P. Kohar, Lars Greve, Tom K. Eller, Daniel S. Connolly, and Kaan Inal. A machine learning framework for accelerating the design process using CAE simulations: An application to finite element analysis in structural crashworthiness. *Computer Methods in Applied Mechanics and Engineering*, 385:114008, 2021.

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.

Yves Le Guennec, Jean-Patrick Brunet, Fatima-Zohra Daim, Ming Chau, and Yves Tourbier. A parametric and non-intrusive reduced order model of car crash simulation. *Computer Methods in Applied Mechanics and Engineering*, 338:186–207, 2018.

Seokki Lee, Min-Chul Park, Giyong Hong, and Changwook Jeong. Buckingham π-invariant test-time projection for robust PDE surrogate modeling. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d6f1710becf448dc79516dbbf33c4a79-Abstract-Conference.html.

Haoran Li, Tobias Lehrer, Yingxue Zhao, Haosu Zhou, Philipp Stocker, Tobias Pfaff, Marcus Wagner, and Nan Li. Mask-Morph Graph U-Net: A generalisable mesh-based surrogate for crashworthiness field prediction under large geometric variation. *arXiv preprint arXiv:2605.15231*, 2026a.

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, 2026b. arXiv:2503.17386.

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.

Ning Liu, Lu Zhang, Tian Gao, and Yue Yu. Disentangled representation learning for parametric partial differential equations. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ef52dae24a9b4a228aedef3b66454123-Abstract-Conference.html.

Sheng Liu, Conghao Liu, Xunan An, Xin Liu, and Liang Hao. Intelligent damage prediction during vehicle collisions based on simulation datasets. *Inventions*, 10(3):40, 2025.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022.

Luminary Cloud. SHIFT-Crash: High-fidelity crash simulation dataset for automotive structural mechanics. Hugging Face dataset repository, 2026. URL https://huggingface.co/datasets/luminary-shift/SHIFT-Crash. SHIFT Models initiative; CC-BY-NC-4.0.

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.

Luis Mandl, Somdatta Goswami, Lena Lambers, and Tim Ricken. Separable physics-informed DeepONet: Breaking the curse of dimensionality in physics-informed machine learning. *Computer Methods in Applied Mechanics and Engineering*, 434:117586, 2025.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for spatiotemporal surrogate models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 119301–119335, 2024.

Michael D. McKay, Richard J. Beckman, and William J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.

Mohammad Amin Nabian, Sudeep Chavare, Deepak Akhare, Rishikesh Ranade, Ram Cherukuri, and Srinivas Tadepalli. Automotive crash dynamics modeling accelerated with machine learning. In *WCX SAE World Congress Experience*, number 2026-01-0568 in SAE Technical Paper. SAE International, 2026. arXiv:2510.15201.

National Highway Traffic Safety Administration. Crash simulation vehicle models. Official model repository, 2026. URL https://www.nhtsa.gov/crash-simulation-vehicle-models. Accessed: 2026-05-04.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, et al. The Well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 44989–45037, 2024.

OpenRadioss. OpenRadioss. GitHub repository, 2026. URL https://github.com/OpenRadioss/OpenRadioss. Accessed: 2026-05-04.

Ying Pang, Jingyuan Wang, Jiahao Ji, and Fanhao Mu. OrthoSolver: A neural proper orthogonal decomposition solver for PDEs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/4515d3012ee9a54cd008aed29f0fc7fa-Abstract-Conference.html.

Rafael Pastrana, Eder Medina, Isabel M. de Oliveira, Sigrid Adriaenssens, and Ryan P. Adams. Real-time design of architectural structures with differentiable mechanics and neural networks. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a421e88bb7db67cd0401475a0ed4a17d-Abstract-Conference.html.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018.

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.

Afsal Pulikkathodi, Ludovic Chamoin, Elisabeth Lacazedieu, Juan Pedro Berro Ramirez, Laurent Rota, and Malek Zarroug. Nonintrusive local/global coupling with local deep learning-based models for the effective simulation of spotwelded structures under impact. *International Journal for Numerical Methods in Engineering*, 126(14):e70086, 2025.

Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92 of *UNITEXT*. Springer, 2016.

Mansi Sakarvadia, Kareem Hegazy, Amin Totounferoush, Kyle Chard, Yaoqing Yang, Ian Foster, and Michael W. Mahoney. The false promise of zero-shot super-resolution in machine-learned operators. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/71d1fdbaa2c57c308e3abb27cbb491c8-Abstract-Conference.html.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. 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.

Zeyuan Song and Zheyu Jiang. Adaptive Mamba neural operators. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/dfd1da26fd06a73e4561ecea7d44a97e-Abstract-Conference.html.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71242–71262, 2023.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611, 2022.

Shilong Tao, Zhe Feng, Shaohan Chen, Weichen Zhang, Zhanxing Zhu, and Yunhuai Liu. Neural latent arbitrary Lagrangian-Eulerian grids for fluid-solid interaction. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7feb98f84ee34f697944333d4faec0bd-Abstract-Conference.html.

Simon Thel, Lars Greve, Bram van de Weg, and Patrick van der Smagt. Introducing finite element method integrated networks (FEMIN). *Computer Methods in Applied Mechanics and Engineering*, 427:117073, 2024.

Simon Thel, Lars Greve, Maximilian Karl, and Patrick van der Smagt. Accelerating crash simulations with finite element method integrated networks (FEMIN): Comparing two approaches to replace large portions of a FEM simulation. *Computer Methods in Applied Mechanics and Engineering*, 443:118046, 2025.

Luke Thompson, Davy Guan, Slade Matthews, Dai Shi, Junbin Gao, and Andi Han. ATOM: A pretrained neural operator for multitask molecular dynamics. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8adbebb54e63b5f82b306847861b9168-Abstract-Conference.html.

Winfried van den Dool, Maksim Zhdanov, Yuki M. Asano, and Max Welling. Adaptive mesh quantization for neural PDE solvers. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=NN17y897WG. Presented at ICLR 2026, Journal Track.

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

Nils Wandel, Stefan Schulz, and Reinhard Klein. Metamizer: A versatile neural optimizer for fast and accurate physics simulations. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/911dd89c81efc624c4e1c39381179505-Abstract-Conference.html.

Hao Wang, Yu Liu, Daniel Biggs, Haoru Wang, Jiandong Yu, and Ping Huang. Learning deformable body interactions with adaptive spatial tokenization. *Transactions on Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=qBOEHzkr1P. Presented at ICLR 2026, Journal Track.

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

Xiaogang Wang, Hongyu Wu, Wenfeng Song, and Kai Xu. Fracture-GS: Dynamic fracture simulation with physics-integrated Gaussian splatting. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3346f8d7c518513060e6ba087e0d5b7e-Abstract-Conference.html.

Shizheng Wen, Arsh Kumbhat, Levi Lingsch, Sepehr Mousavi, Yizhou Zhao, Praveen Chandrashekar, and Siddhartha Mishra. Geometry aware operator transformer as an efficient and accurate neural surrogate for PDEs on arbitrary domains. In *Advances in Neural Information Processing Systems*, volume 38, pp. 155423–155501, 2025.

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.

Xinling Yu, Sean Hooten, Ziyue Liu, Yequan Zhao, Marco Fiorentino, Thomas Van Vaerenbergh, and Zheng Zhang. Separable operator networks. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=RYtJmFDAxv.

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

Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: PDE-conditional transformers towards universal neural PDE solvers. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 79061–79088. PMLR, 2025.

APPENDIX CONTENTS

A Model configuration and cost **18**

Hardware and computational cost. . . . 18

Model configuration. . . . 18

Training procedure. . . . 18

Classical reduced-order model. . . . 18

B Detailed results per dataset **18**

B.1 Classical reduced-order modeling on fixed and morphing meshes . . . 18

B.2 Combining classical and neural predictions . . . 19

C Contributions of architecture and pretraining **19**

C.1 Joint versus single-dataset training. . . . 19

C.2 Adaptation to an unseen vehicle: fine-tuning versus training from scratch . . . 20

D Full-mesh evaluation **21**

E Use cases **21**

Design sweeps on a fixed mesh. . . . 21

Prediction from morphed geometry with parameters withheld. . . . 22

F Ablations and controls **22**

F.1 Temporal rank . . . 22

F.2 Full-trajectory prediction versus autoregressive rollout . . . 23

Alternative one-shot decoders and held-out frames. . . . 23

F.3 Anchor tokens: count and content . . . 24

F.4 Residual learning with a classical surrogate . . . 25

G Evaluation protocol and baseline details **25**

Metric. . . . 25

Simulation-level uncertainty. . . . 26

H The Yaris-yield dataset **26**

Motivation for material-dependent prediction. . . . 26

Dataset construction. . . . 26

Effect of yield-strength conditioning. . . . 27

I Full-field deformation predictions across vehicles **27**

A Model configuration and cost

Hardware and computational cost. We train each ImpactFormer network on a single NVIDIA A100-80GB GPU in approximately 1.7 GPU-hours. On the same GPU, ImpactFormer predicts the complete displacement trajectory of one simulation in 2.7 s on the full 1,067,287-node Neon mesh (16 frames), 2.3 s on the 916,742-node Silverado mesh (40 frames) and 0.9 to 1.0 s on the Yaris and SHIFT-Crash meshes, with a peak GPU memory of at most 2.1 GB (bf16, queries in chunks of 65,536 nodes, data loading excluded); on the benchmark node sets it takes 0.57 to 0.77 s. The classical reduced-order model is fitted separately for each vehicle in minutes on a CPU, using an $n_{\text{train}} \times n_{\text{train}}$ Gram matrix.

Model configuration. ImpactFormer has 19.2M parameters and an attention backbone with a width of 384 and eight attention blocks (Vaswani et al., 2017). The input combines variable-length parameter tokens with 3,072 anchor tokens during training and 2,048 during evaluation. Training anchors are resampled at each step. Dataset-specific FiLM modulation (Perez et al., 2018) conditions the token embeddings, and query cross-attention produces the spatial coefficients of a shared temporal basis $b(\tau)$ of size $R = 32$. Query nodes carry the same thickness channel as the anchors. The temporal basis b is a multilayer perceptron with two hidden layers of 128 units. The FiLM scale and shift are computed from a learned dataset embedding and from the mean of the parameter-token embeddings, and are applied to the anchor embeddings before the trunk and to the query embeddings before cross-attention. Coordinates are centered on the bounding box of the undeformed structure and divided by half its diagonal; displacements are divided by the same length, and predictions are rescaled to millimeters.

Training procedure. The model is trained using the mean squared error of normalized nodal displacement, evaluated at a randomly sampled subset of query nodes for each simulation. To balance the datasets, SHIFT-Crash is sampled with a probability of approximately one third, while each of the four smaller datasets is sampled with a probability between 0.14 and 0.20. The optimizer, training schedule, and sampling probabilities are selected using validation data; the aggregate validation score used for every selection, and reported in the rank sweep of Appendix F.1, is the mean over the five datasets of the per-dataset validation RMSE. Ablations retain these settings unless otherwise stated and use paired random seeds.

Classical reduced-order model. CrashDoE represents the training displacement trajectories using proper orthogonal decomposition (POD) (Berkooz et al., 1993; Quarteroni et al., 2016) and learns a regularized regression from the design parameters to the POD coefficients. Its hyperparameters are selected using validation data. For the hybrid comparisons, the classical model is refitted out of fold.

B Detailed results per dataset

In this section, we provide additional results for the comparison with classical reduced-order modeling and for the combination of ImpactFormer with classical predictions. These experiments extend Sections 5.2 and 5.4. All results are test RMSE values in millimeters, computed for each simulation and then averaged over the test set using the fixed evaluation nodes defined in Appendix G. For the four datasets with neural baselines, the baseline reference is the lowest test RMSE among three runs of the strongest neural solver, CrashSolver. The separate Yaris-yield reference is the mean error of our single-dataset model over three seeds (Appendix H).

B.1 Classical reduced-order modeling on fixed and morphing meshes

To examine the performance of classical reduced-order modeling on fixed and morphing meshes, we apply the same POD-plus-regression procedure to both settings (Table 3). The validation search grid is unchanged, and each dataset is evaluated on its fixed benchmark node set. In SHIFT-Crash, the geometry changes between simulations but the node topology is shared, allowing displacement trajectories to be represented in a common POD basis. The surrogate receives the six design parameters rather than the geometry itself. Increasing the training set from 800 to 4,088 simulations reduces its RMSE by 0.19 mm, while quadrupling the number of POD modes reduces RMSE by 0.013 mm. These changes do not close the performance gap on SHIFT-Crash.

B.2 Combining classical and neural predictions

We combine the classical prediction E_0 with the ImpactFormer prediction $\hat{u}$ using the convex combination $w\,E_0 + (1-w)\,\hat{u}$. Table 6 reports the test results, with the weight w selected separately for each vehicle using validation data (Figure 3). The selected weights place greater emphasis on the classical model for Neon and assign equal weight to the two components for Yaris. We do not evaluate a hybrid on SHIFT-Crash: its classical model is fitted on the 229,722-node common evaluation set only and receives the six design parameters but no geometry, so a blend would exist on those nodes alone and could not extend to the remaining 146,871 nodes of a morphed mesh without a refit.

Table 6: Combining classical and ImpactFormer predictions on fixed meshes. Values are mean per-simulation test RMSE (mm; lower is better). The weight w assigned to the classical field E_0 is selected separately for each dataset using validation data, and the classical model is refitted out of fold. The last column reports the relative change from the baseline reference, which on SHIFT-Crash was trained on a different node subset (Appendix G); there, CrashSolver retrained on the full mesh scores 6.088 mm on these nodes (Table 2), and ImpactFormer is 0.9% below it. Bold: lowest entry of a row.

dataset	baseline	classical alone	ImpactFormer	hybrid (w)	vs. baseline
Neon	3.959	2.845	3.481	**2.814** (w=0.9)	-28.9%
Yaris	6.026	5.982	5.481	**5.404** (w=0.5)	-10.3%
Silverado	8.171	7.322	7.521	**6.420** (w=0.7)	-21.4%
SHIFT-Crash	6.239	—	**6.031**	—	-3.3%

C Contributions of architecture and pretraining

The main benchmark compares ImpactFormer with neural solvers that differ in architecture and training data. We conduct two additional experiments to examine these contributions separately: training the same architecture on individual datasets (Appendix C.1), and comparing fine-tuning with training from scratch on identical simulations (Appendix C.2). Unless stated otherwise, results are mean test RMSE values in millimeters over three random seeds, using the fixed evaluation node sets.

C.1 Joint versus single-dataset training

ImpactFormer combines a high-performing crash-prediction architecture with the ability to learn across vehicles, meshes, and design parameterizations. To examine these contributions, we compare a model trained jointly on all five datasets with separate models trained on each dataset, keeping the architecture unchanged.

We consider two training budgets for the separate models. With *full exposure*, each model receives the same total number of optimizer steps as the joint model, but uses all updates on its own dataset. With *matched exposure*, it receives the same expected number of updates on that dataset as the joint model.

Table 7: Comparison of single-dataset and joint training using test RMSE (mm; lower is better). Full exposure matches the total optimizer steps of joint training; matched exposure matches the expected updates on each dataset. We report mean $\pm$ standard deviation over three seeds for single-dataset ImpactFormer and CrashSolver. Bold: lowest entry of a row.

	ImpactFormer, one dataset			CrashSolver
Dataset	Full exposure	Matched exposure	ImpactFormer, joint	Mean of 3
Neon	3.547 ± 0.172	5.446 ± 2.066	**3.481**	3.974 ± 0.017
Yaris	6.116 ± 0.091	5.992 ± 0.129	**5.481**	6.151 ± 0.119
Silverado	12.841 ± 2.769	9.830 ± 0.845	**7.521**	8.286 ± 0.108
SHIFT-Crash	**5.904** ± 0.034	6.033 ± 0.164	6.031	6.341 ± 0.102

As shown in Table 7, the ImpactFormer architecture is highly accurate even when trained on individual datasets. Under full exposure, the separate models achieve mean RMSE values of 3.55 mm on Neon and 5.90 mm on SHIFT-Crash, improving on the three-seed CrashSolver mean by approximately 11% and 7%, respectively. This shows that the performance gains do not depend solely on access to additional datasets.

Joint training is more accurate than matched-exposure single-dataset training on all four datasets, and than full-exposure single-dataset training on Neon, Yaris and Silverado; on Silverado, it also avoids the instability of single-dataset training. SHIFT-Crash, the largest dataset, favors full-exposure single-dataset training. The results therefore identify where sharing training data improves accuracy and where a single-dataset model retains an advantage.

Beyond these dataset-specific comparisons, joint training enables one network to predict across all five datasets using a single set of weights. It also provides pretrained weights that can be reused when introducing a new vehicle. As shown in Appendix C.2, fine-tuning on the held-out Silverado achieves lower error than training the same architecture from scratch at matched training budgets. Together, these results demonstrate the two complementary strengths of ImpactFormer: an architecture that delivers strong accuracy on individual datasets, and a shared model that predicts all five datasets with one set of weights and adapts to a new vehicle more data-efficiently than the same network trained from scratch.

C.2 Adaptation to an unseen vehicle: fine-tuning versus training from scratch

We exclude Silverado from pretraining to evaluate transfer from sedans and morphed Yaris geometries to a pickup truck. Without adaptation, ImpactFormer has a mean RMSE of 135.2 mm across three random seeds (range: 123.5–143.6 mm). Zero-shot, the withheld vehicle receives a fresh all-zero dataset row, whose modulation is not the identity after training; its three parameter names are shared with Neon and Yaris. Using the Yaris dataset embedding lowers RMSE to 124.5 mm, but the error remains high. Fine-tuning on 4, 8, and 16 Silverado simulations reduces RMSE to 26.4 ± 0.5, 20.1 ± 0.3, and 16.2 ± 0.7 mm, respectively, 80%, 85%, and 88% below zero-shot prediction, with a short, fixed training budget (Table 5). ImpactFormer, trained jointly with all 56 Silverado training simulations, reaches 7.52 mm, so sixteen simulations, 29% of the Silverado training set, already deliver 93% of the error reduction of full training.

We compare fine-tuning the pretrained model with training the same architecture from scratch on identical Silverado simulations (Table 8). Paired seeds use the same training subsets and the same four validation simulations. Both initializations are evaluated using the short adaptation budget of Table 5 and a budget about three times longer. Under the short budget the models trained from scratch use the adaptation settings; under the long budget they use the optimizer settings of the final configuration, and we also report their validation-selected epochs.

Table 8: Fine-tuning versus training from scratch on the same Silverado simulations. Values are test RMSE (mm), reported as mean $\pm$ standard deviation over three paired seeds. The short budget is the adaptation budget used in Table 5; the long budget uses about three times as many steps. Models trained from scratch are also evaluated at their best validation epoch under the long budget. The last row gives reference results using all 56 Silverado training simulations. All columns use the same training subset for every seed (the first K runs of the split, for K Silverado runs) and the same four validation runs; the classical ROM is fitted to that subset with hyperparameters selected on the validation runs.

	pretrained, fine-tuned		from scratch			classical
Silverado runs	short budget	long budget	short budget	long budget	long, best val. epoch	ROM
4	26.368 ± 0.500	24.686 ± 0.261	66.625 ± 0.960	51.373 ± 4.828	51.241 ± 7.291	21.547
8	20.115 ± 0.252	18.137 ± 0.271	57.081 ± 1.428	43.545 ± 9.798	43.695 ± 10.791	17.511
16	16.182 ± 0.664	13.253 ± 0.691	69.875 ± 2.173	18.194 ± 1.884	15.479 ± 0.586	11.675
56 (all)	joint training: 7.521		single-dataset ImpactFormer, full exposure: 12.841 ± 2.769			7.322

Fine-tuning achieves lower error than training from scratch at matched budgets. Under the longer budget, the advantage is largest when only four or eight simulations are available. Increasing the training budget improves the models trained from scratch, particularly with sixteen simulations, but

also improves the fine-tuned models. The approximately eleven-minute adaptation time in Table 5 therefore depends on starting from pretrained weights; the same short budget is insufficient for training from scratch. On this fixed-mesh vehicle, however, the classical ROM fitted to the same runs is more accurate than every neural variant, significantly so against the short-budget fine-tune with four and sixteen runs (paired Wilcoxon p=0.004 and 0.003 over the 15 test runs) and within the uncertainty against the long budget (p=0.07, 0.30, 0.21). Its error also depends strongly on which runs are used: over 100 random subsets it reaches 25.7 ± 5.7, 17.3 ± 2.3 and 13.3 ± 1.7 mm, whereas the variation of the fine-tuned model across subsets was not measured. Pretraining is therefore the better route among neural models, and the classical model remains the stronger few-shot surrogate when the mesh and parameterization of the new vehicle are fixed.

D Full-mesh evaluation

We evaluate displacement predictions on the complete structural mesh to examine performance beyond the fixed benchmark node subset. ImpactFormer samples training queries from the full mesh and can predict at any node, while the neural baselines in the standard benchmark are trained on a node subset of the benchmark's size. Table 9 compares the subset and full-mesh results for ImpactFormer, the classical reduced-order model, and their hybrid. ImpactFormer scales to the complete meshes without retraining, including the 1,067,287-node Neon, and on the three vehicles it is 11–15% more accurate on the full mesh than CrashSolver retrained at full resolution (Table 2). Anchor selection is unchanged from the benchmark protocol, and the subset results reproduce Table 2. AB-UPT, like ImpactFormer, decodes at arbitrary query points and could be trained on full-mesh queries; we did not retrain it in this way.

Table 9: Comparison of test RMSE (mm) on the fixed benchmark node set and the full structural mesh of the same test simulations. The hybrid combines ImpactFormer and the classical prediction using validation-selected weights (Table 6). The SHIFT-Crash classical model is evaluated only on the common node set. Yaris-yield is omitted because its benchmark already uses the full mesh.

dataset	nodes scored	ImpactFormer	classical ROM	hybrid
Neon	pinned, 296,171	3.481	2.845	2.814
	full mesh, 1,067,287	6.308	5.764	5.702
Yaris	pinned, 210,621	5.481	5.982	5.404
	full mesh, 345,278	5.561	5.939	5.435
Silverado	pinned, 254,422	7.521	7.322	6.420
	full mesh, 916,742	7.662	7.622	6.726
SHIFT-Crash	common set, 229,722	6.031	6.706	–
	full mesh, 376,593	6.892	–	–

E Use cases

We examine applications to the vehicle design process (Kohar et al., 2021): design sweeps on a fixed mesh and prediction from morphed geometry with the geometric parameters withheld; adaptation to an unseen vehicle is analyzed in Appendix C.2. Figure 5 summarizes the adaptation results for Silverado, the distribution of errors across its test simulations, and the error over time on Neon. On Neon, the error of both models grows as the front structure crushes, and ImpactFormer is more accurate than CrashSolver at every predicted frame: its error is about one third of CrashSolver's at the first frame (0.34 versus 1.08 mm) and 4% lower at the final frame (6.88 versus 7.14 mm), so its advantage is largest early in the impact (Figure 5c).

Design sweeps on a fixed mesh. The Neon example belongs to a three-parameter design-of-experiments study on a fixed mesh. Evaluating a new candidate with the finite-element solver OpenRadioss (OpenRadioss, 2026) requires approximately 95 core-hours (Elrefaie et al., 2026), whereas ImpactFormer predicts the complete displacement trajectory in one forward pass per network.

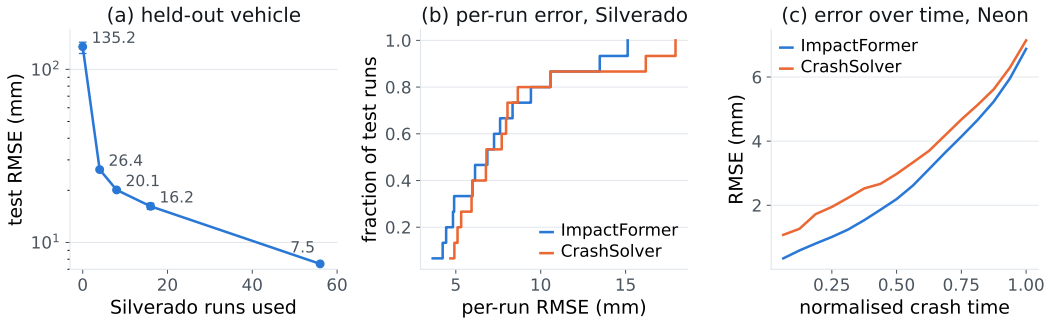

Figure 5: Adaptation and prediction accuracy in vehicle design applications. (a) Silverado test RMSE after adaptation with increasing numbers of simulations, shown on a logarithmic scale. Results are averaged over three seeds; error bars show standard deviation for adaptation and the range for zero-shot inference. RMSE decreases from 135.2 mm without adaptation to 16.2 mm with sixteen simulations, compared with 7.52 mm for ImpactFormer trained jointly with all 56 simulations. (b) Distribution of per-simulation RMSE on the 15 Silverado test runs for ImpactFormer and CrashSolver: paired run by run, ImpactFormer is more accurate on 11 of the 15 runs (Appendix G). (c) RMSE over normalized crash time, averaged over the Neon test runs: ImpactFormer is more accurate throughout the crash.

On the three fixed-mesh vehicles, combining ImpactFormer with the vehicle-specific classical model reduces RMSE by 10–29% relative to the strongest neural baseline and by 1.1–12.3% relative to the better of its two components (Table 6).

Prediction from morphed geometry with parameters withheld. SHIFT-Crash allows us to evaluate whether geometric changes can be inferred without explicitly providing the morphing parameters. Its front-rail length, cabin length, and roof height vary by -40 to $+100$ mm, and its width by -40 to $+40$ mm, across simulations. ImpactFormer can read these changes directly from the mesh: for a variant trained with parameter-token dropout, withholding all four morph parameters increases RMSE by only 5.5%, from 7.25 to 7.65 mm, while replacing its anchors with those from a different simulation raises RMSE to 18.5 mm, which shows that the model recovers each morph from the anchor geometry. Parameter-driven surrogates such as the classical ROM cannot do this by construction, since without the parameters they receive no information about the design. Geometry also cannot replace parameters that leave the mesh unchanged: on Neon, withholding the parameter tokens substantially increases mean test RMSE because the fixed mesh does not reveal the shell-thickness changes.

F Ablations and Controls

F.1 Temporal Rank

Table 10 shows the temporal-rank sweep across all five datasets, using one random seed per rank and keeping all other training settings fixed. Although validation error shows diminishing improvements around $R = 8$, increasing the rank to $R = 32$ still reduces test error on every dataset. For reference, an uncentered temporal POD fitted to 48 training runs per dataset reconstructs the test trajectories (48 of the 504 on SHIFT-Crash) on the benchmark nodes with an error of 1.6, 5.4, 7.0, 3.3 and 5.0 mm at four modes and 0.5, 2.2, 2.0, 1.3 and 2.0 mm at eight (Neon, Yaris, Silverado, SHIFT-Crash, Yaris-yield), even though four modes hold more than 99.9% of the trajectory energy; a single basis shared by all five datasets is at most about 1.3 mm worse on any dataset at four modes, depending on how the datasets are weighted in the fit, and at most 0.1 mm worse at eight. On four of the five datasets, most of the cost of $R{=}4$ is therefore this representation floor, not a deficiency of the shared dictionary (on Neon, less than half); because the predicted trajectory of one run has rank at most $\min(R, T{-}1)$, values of R above 20 matter only through the Silverado grid and through optimization.

Table 10: Effect of temporal rank R (one random seed per rank, all other settings as in the final configuration): change in mean per-simulation test RMSE for each dataset and in the aggregate validation score, relative to the final configuration with $R = 32$, whose values (mm) are given in the last row; lower is better.

R	val. macro ↓	Neon ↓	Yaris ↓	Silverado ↓	SHIFT-Crash ↓	Yaris-yield ↓
2	+137.0%	+213.6%	+199.5%	+145.0%	+114.0%	+82.0%
4	+30.4%	+30.5%	+38.9%	+54.7%	+13.2%	+13.1%
8	+3.2%	+8.5%	+7.0%	+13.0%	+0.8%	+10.6%
16	+3.4%	+5.2%	+2.6%	+8.8%	+3.9%	+7.2%
32 (mm)	6.724	3.772	5.969	8.399	6.319	9.475

F.2 Full-trajectory prediction versus autoregressive rollout

We extend the experiments in Section 5.3 to examine how the temporal representation affects prediction accuracy while keeping the attention backbone unchanged. Figure 6 presents the comparison with autoregressive rollout alongside the temporal-rank study. Predicting complete trajectories through the shared temporal basis achieves the lowest error among the temporal schemes tested. Across the four benchmark datasets, the relative increase in RMSE from autoregressive rollout is larger for datasets with more predicted frames (Spearman $\rho = 1.000$, Pearson $r = 0.985$), with a fitted slope of approximately 1.8 percentage points per additional frame. This trend is observed across datasets with different trajectory lengths, rather than by varying the rollout length within a single dataset.

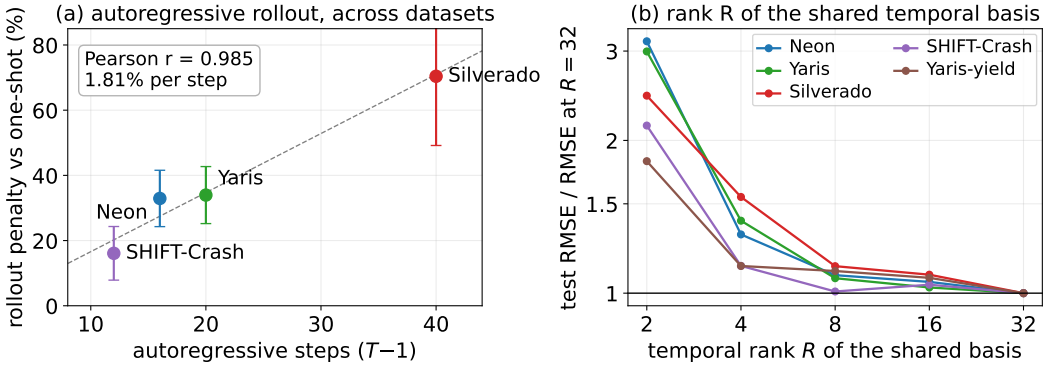

Figure 6: Effect of temporal prediction strategy and basis rank. (a) Relative increase in RMSE from autoregressive rollout as a function of the number of predicted frames, with results from three seeds and a linear fit. (b) Test RMSE as a function of temporal rank R, normalized by each dataset's result at $R = 32$. Both experiments retain the same attention backbone.

Table 11 reports the results for three paired seeds. Yaris-yield falls below the trend of Figure 6a and has the widest spread across seeds relative to its mean penalty. The experiment matches optimizer steps rather than computational cost.

Alternative one-shot decoders and held-out frames. We compare the shared temporal basis with two alternative non-autoregressive decoders to examine its contribution to prediction accuracy and computational cost. As shown in Table 12, a *time-conditioned decoder* includes τ in each query and predicts displacement separately at each (x, τ), while a *direct multi-frame head* outputs all $3(T-1)$ displacement components of a node without a temporal basis. Both variants use the same three random seeds as the shared-basis model. The time-conditioned decoder achieves comparable accuracy on all datasets but requires $T-1$ decoder evaluations per node rather than one. The direct head has higher mean error on every dataset, although not in every seed; one training run reaches 11.9 mm on Neon and 15.9 mm on Silverado. The shared basis therefore combines the accuracy of time-conditioned prediction with the efficiency of decoding each node only once.

Table 11: Comparison of full-trajectory prediction and autoregressive rollout using the same backbone and three paired random seeds. Relative increases in test RMSE are computed against the corresponding full-trajectory ImpactFormer run trained with the same seed and then averaged across seeds. We also report the individual relative increases, the number of seeds with higher rollout error, and the number of autoregressive prediction steps ($T-1$). Only the temporal prediction scheme differs.

	Neon	Yaris	Silverado	SHIFT-Crash	Yaris-yield
paired rollout penalty	+32.9%	+34.0%	+70.4%	+16.1%	+10.4%
rollout worse on	3/3	3/3	3/3	3/3	3/3
per-seed penalty (%)	35.6/39.9/23.3	29.3/44.0/28.6	85.7/79.3/46.2	13.8/25.2/9.3	12.4/18.9/0.1
autoregressive steps ($T-1$)	16	20	40	12	20

We also examine whether the continuous temporal representation supports prediction at times not included in training. When trained and validated on alternate frames and evaluated on all frames, the factorized model shows substantially higher errors than when supervision is provided at every frame, indicating poor interpolation to unsupervised time points under this training setup.

The model trained on all output frames also shows higher errors between those frames. On five test simulations per dataset, evaluation at temporal midpoints gives RMSE values of 28.4, 8.3, 18.3, 9.1, and 9.6 mm against interpolated references, compared with 3.5, 6.4, 5.5, 6.8, and 8.4 mm at the original output frames, in the same dataset order. Thus, although the basis can be evaluated continuously in time, its demonstrated accuracy is concentrated on the supervised output grids.

The learned basis does not explicitly enforce zero initial displacement. Evaluating the decoder at $\tau = 0$ on three test simulations per dataset produces RMS displacements of 60–146 mm. The condition $u(x, 0) = 0$ therefore follows from the definition of the undeformed initial state rather than from the decoder construction. We have not evaluated a fixed temporal basis, such as the POD modes discussed in Appendix F.1, as an alternative to the learned basis.

Table 12: Comparison of temporal decoders using the same attention trunk and three paired random seeds. Values are percentage changes in test RMSE relative to ImpactFormer's shared-basis decoder ($R = 32$), computed against the corresponding reference run trained with the same seed and then averaged across seeds. Positive values indicate higher error than the shared-basis reference. Decoder passes per query node are reported for $T-1$ predicted frames.

decoder	passes per node	Neon	Yaris	Silverado	SHIFT-Crash	Yaris-yield
time-conditioned decoder	$T-1$	+1.3%	−0.6%	+4.9%	+0.5%	−4.3%
direct multi-frame head	1	+70.9%	+3.8%	+22.7%	+4.6%	+5.9%
autoregressive rollout	$T-1$	+32.9%	+34.0%	+70.4%	+16.1%	+10.4%

F.3 Anchor tokens: count and content

Anchor tokens encode geometry for the attention backbone but do not determine the output resolution. The decoder queries each output node using its coordinates, allowing prediction at all 210,621–296,171 evaluation nodes of the four datasets with neural baselines even when few anchors are used. For the same reason, one trained model predicts on complete meshes of more than one million nodes (the 1,067,287-node Neon; Appendix D) from only 2,048 anchors, without retraining. We retrain the final configuration with 64, 256, 1,024, and 8,192 anchors, using the same three random seeds as the default configuration with 3,072 anchors (Table 13). Most differences at 256 and 1,024 anchors fall within the variation across seeds. Increasing the count to 8,192 does not improve accuracy on any dataset and takes 2.3 times as long to train.

On 40 SHIFT-Crash test runs, swapping anchors between morphed geometries barely changes the RMSE of an ImpactFormer network (6.033 to 6.028 mm), while collapsing all anchors to one point raises it to 46.6 mm. This suggests that the model uses anchors mainly for geometric context rather than to distinguish individual morphs. The earlier parameter-dropout variant (Appendix E) behaves

Table 13: Effect of anchor count on test RMSE relative to the default ImpactFormer configuration, using three paired random seeds. Percentage changes are computed against the corresponding default run trained with the same seed and then averaged across seeds. Each ablated configuration uses the stated number of anchors during both training and evaluation, whereas the default uses 3,072 training anchors and 2,048 evaluation anchors. The standard deviation of these relative changes ranges from 1 to 12 percentage points across configurations and datasets.

anchors	Neon	Yaris	Silverado	SHIFT-Crash	Yaris-yield
64	+11.5%	+4.0%	+6.9%	+4.8%	+0.4%
256	+4.4%	+0.2%	+1.8%	−0.4%	+5.3%
1,024	+1.2%	+1.4%	+4.3%	−0.5%	+2.2%
8,192	+5.2%	+0.9%	+15.4%	+0.6%	+2.9%

differently. With the four morph parameters withheld, swapping anchors increases RMSE from 7.65 to 18.50 mm, while reducing the anchor count from 2,048 to 64 increases it from 7.65 to 16.70 mm.

Adding parameter dropout under the final training configuration did not reproduce this behavior. Across three seeds and all 504 SHIFT-Crash test runs, models with 3,072 and 64 anchors achieve 15.57 and 16.09 mm without the morph parameters. Swapping anchors changes these errors only slightly, to 16.33 and 16.10 mm, respectively. The earlier variant was trained three times longer and sampled SHIFT-Crash about twice as often, giving roughly six times the exposure to this dataset.

F.4 Residual learning with a classical surrogate

As an alternative to the hybrid in Appendix B.2, we investigate whether the classical prediction can improve accuracy when used during training rather than combined with the neural prediction afterward. The operator receives the classical field E_0 as an additive prediction and learns the remaining residual. For each training simulation, the classical prediction is fitted using folds that exclude that simulation, avoiding an in-sample fit to the target trajectory.

Across six paired seeds, residual learning reduces the operator's RMSE from 4.105 to 3.381 mm on Neon and from 8.958 to 8.091 mm on Silverado. However, these results remain above the classical model's errors of 2.845 and 7.322 mm, respectively. Scaling the learned residual using a validation-selected factor provides only small improvements over the classical model: 0.000, 0.016, and 0.038 mm on Neon, Yaris, and Silverado. By comparison, combining the independently trained ImpactFormer with the classical prediction reduces error by 0.031, 0.578, and 0.902 mm (Table 6). An operator fitted to the residual of a classical fit inherits the error structure of that fit, so a combination of the two has little left to cancel, whereas an independently trained operator makes different errors. The hybrid therefore combines the two models after training.

G Evaluation protocol and baseline details

Metric. We evaluate displacement prediction accuracy using the root mean square error (RMSE), expressed in millimeters. For each test simulation, RMSE is computed over the three displacement components, the fixed evaluation node set, and all frames except the initial frame. Following the evaluation protocol in Section 3, we report the mean of the per-simulation errors:

$$\mathrm{RMSE}_i = \sqrt{\frac{1}{3\,(T-1)\,|\mathcal{V}|}\sum_{j=1}^{T-1}\sum_{n\in\mathcal{V}}\left\|\hat{u}_i(x_n,t_j)-u_i(x_n,t_j)\right\|_2^2},$$
$$\overline{\mathrm{RMSE}} = \frac{1}{N_{\mathrm{test}}}\sum_{i=1}^{N_{\mathrm{test}}}\mathrm{RMSE}_i\,, \tag{2}$$

where $u_i(x_n,t_j)\in\mathbb{R}^3$ is the ground-truth displacement of node n at the time t_j of output frame j in simulation i, and $\hat{u}_i(x_n,t_j)$ is the corresponding prediction, both in millimeters. Here, T is the number of exported frames including the initial frame $j{=}0$, $\mathcal{V}$ is the fixed evaluation node set, and N_{test} is the number of test simulations.

Simulation-level uncertainty. Table 14 pairs the per-simulation test RMSE of ImpactFormer with that of CrashSolver on the benchmark node sets. ImpactFormer's advantage is not driven by a few favorable simulations: it is more accurate on 88% of the Neon, 73% of the Yaris and Silverado and 67% of the SHIFT-Crash test runs, and the paired differences are significant on Neon, Yaris and SHIFT-Crash ($p < 10^{-3}$), on SHIFT-Crash also against CrashSolver retrained on the full mesh (p=0.019). On Silverado, whose 15 test designs almost all extrapolate beyond the training designs, the interval is wide, but the mean difference of 0.65 mm again favors ImpactFormer. The intervals resample test simulations with the trained models fixed.

Table 14: Paired per-simulation differences in test RMSE (ImpactFormer minus baseline, mm) with 95% bootstrap intervals over test simulations (10,000 resamples), two-sided Wilcoxon signed-rank p, and the share of the n test simulations on which ImpactFormer is more accurate. Baseline: the CrashSolver reference (best of three seeds), and on SHIFT-Crash also CrashSolver retrained on the full mesh and scored on the common node set.

dataset (n)	baseline	difference [95% interval]	p	ImpactFormer lower
Neon (25)	CrashSolver reference	-0.478 [-0.654, -0.271]	1.9×10^{-4}	88%
Yaris (44)	CrashSolver reference	-0.544 [-0.850, -0.272]	5.6×10^{-4}	73%
Silverado (15)	CrashSolver reference	-0.650 [-2.461, $+1.054$]	0.21	73%
SHIFT-Crash (504)	CrashSolver reference	-0.208 [-0.255, -0.160]	5×10^{-20}	67%
	full-mesh-trained CrashSolver	-0.057 [-0.107, -0.005]	0.019	55%

H THE YARIS-YIELD DATASET

Motivation for material-dependent prediction. Yield strength is a primary crashworthiness design variable: it sets the load at which rails and cross-members begin to fold, and therefore how much energy the crush zone absorbs, the deceleration pulse transmitted to the occupants and the intrusion into the cabin. Engineers change it through the choice of steel grade, for example when replacing mild steel with high-strength steel to save weight, and it also scatters between material batches, so a surrogate that ignores it cannot support material selection or robustness studies. The existing benchmarks provide limited evidence of material-sensitive prediction. The full-vehicle datasets of CarCrashNet (Elrefaie et al., 2026) vary impact velocity and shell thickness while keeping material properties fixed. SHIFT-Crash (Luminary Cloud, 2026) varies front-rail yield strength from 250 to 400 MPa alongside four geometric parameters and rail thickness, but our experiments indicate that the evaluated models make little use of the yield-strength input. For an earlier version of our model, trained jointly on SHIFT-Crash and the three CarCrashNet vehicles, shuffling yield values across test simulations changes RMSE by only 0.02 mm (0.3%). Across eight matched test pairs that differ primarily in yield strength, the correlation between predicted and true changes in the displacement field is 0.03. A counterfactual sweep produces a predicted crush slope of $+0.0007$ mm/MPa, with monotonic behavior in the expected physical direction in 17% of cases. Adding per-node material fields provides little improvement: the permutation test changes RMSE by 0.08 mm and the paired-response correlation is 0.02.

The matched ground-truth pairs show measurable material effects, but these are small relative to the accompanying geometric changes. The evaluated models, trained with a mean squared error objective, do not reliably distinguish this response. We therefore construct Yaris-yield to study material-dependent deformation on a fixed geometry with controlled yield-strength variation.

Dataset construction. Yaris-yield uses the coarse 2010 Toyota Yaris model (Center for Collision Safety and Analysis, 2026) as set up in CarCrashNet (Elrefaie et al., 2026), retaining its 345,278-node topology and 21 frames over 0.20 s. We add two yield-strength scale factors to the original three-parameter design: σ_{fs} for the front-support group and σ_{lr} for the lower-rail and subframe group. Both range from 0.7 to 1.3. Impact velocity ranges from 40 to 65 km/h, and the two thickness scales range from 0.8 to 1.2. Yield scaling multiplies the whole plastic stress–strain curve of the affected materials, initial yield stress and hardening alike, by the scale factor; the materials are cloned for each part group so that other parts remain unchanged.

We generate 120 designs using Latin hypercube sampling (McKay et al., 1979) and simulate them with OpenRadioss (OpenRadioss, 2026). Each simulation takes approximately 40 minutes using 64 MPI ranks. One high-velocity design diverges through shell rupture and is excluded, leaving 119 simulations split into 95 training, 12 validation, and 12 test runs.

In the ground-truth data, displacement-field sensitivity per standard deviation of each parameter is 14.3 mm for velocity, 3.65 mm for lower-rail yield strength, 2.4–2.8 mm for the thickness scales, and 1.45 mm for front-support yield strength. The yield-strength effects are therefore comparable in magnitude to those of thickness variation, providing material contrasts for evaluating the surrogate.

Effect of yield-strength conditioning. On Yaris-yield, including the two yield-strength tokens, together with a per-node material field and the nodal shell thickness, reduces our single-dataset model's test RMSE by 29%, from 14.58 ± 0.98 to 10.41 ± 0.45 mm across three seeds. The tokens also improve the predicted response to material changes. On held-out designs, the correlation between predicted and true displacement responses to changes in yield strength is 0.67 for the front support and 0.88 for the lower rail, compared with 0.09 and -0.52 when the tokens are withheld. Shuffling yield values between test runs increases RMSE by 3.8 mm for the conditioned model, while leaving the model without yield tokens unchanged.

These results show that the architecture can learn material-dependent deformation when the training data include variations in yield strength. We therefore include Yaris-yield as the fifth dataset in joint training. Jointly trained ImpactFormer achieves 8.930 mm, 14% below the 10.41 mm of the single-dataset model. However, the dataset contains only 95 training simulations, and the models capture the direction of the response to yield-strength changes more reliably than its magnitude or shape across counterfactual sweeps.

I FULL-FIELD DEFORMATION PREDICTIONS ACROSS VEHICLES

Figures 7–10 compare ImpactFormer predictions with finite-element results on the full structural meshes of Neon, Yaris, Silverado and SHIFT-Crash, using the test cases in Figure 4. Each figure shows three frames, at about $\tau = 1/3$, $2/3$ and 1, with the exact normalized times indicated in the rows. Columns show the reference deformation in gray, the predicted deformation in blue, and the prediction colored by node-wise error in millimeters. The upper limit of each error scale is the 99th percentile of the final-frame error. Across these examples, ImpactFormer accurately reproduces the deformation sequence of every vehicle: the cabin and rear body remain within a few millimeters of the reference trajectories, and the larger errors are confined to the most strongly deforming parts of the front structure and the wheels.

Figure 7 shows the Neon test run at three instants of the crash.

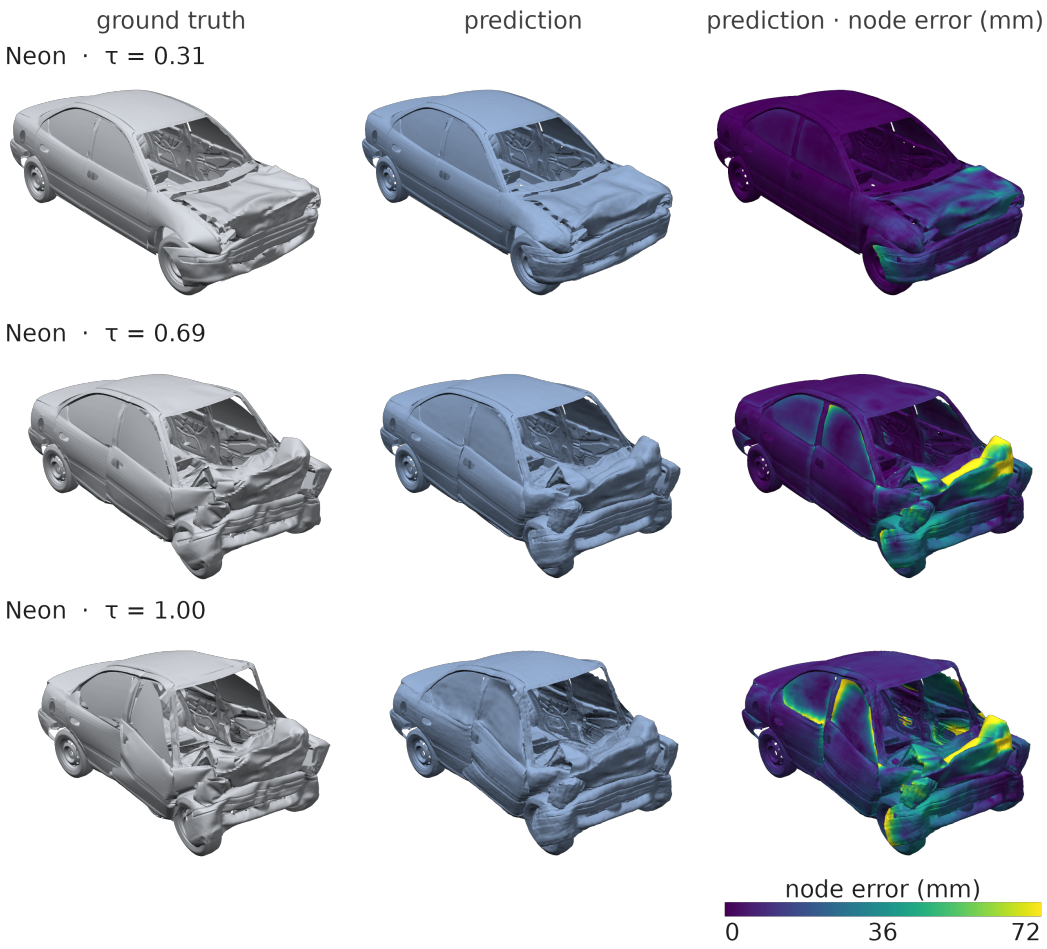

Figure 7: Full-field deformation prediction for a Neon test simulation at three stages of the crash. ImpactFormer achieves an RMSE of 7.139 mm on this run. The node-wise error scale ranges from 0 to 72 mm.

Figure 8 shows the Yaris test run at three instants of the crash.

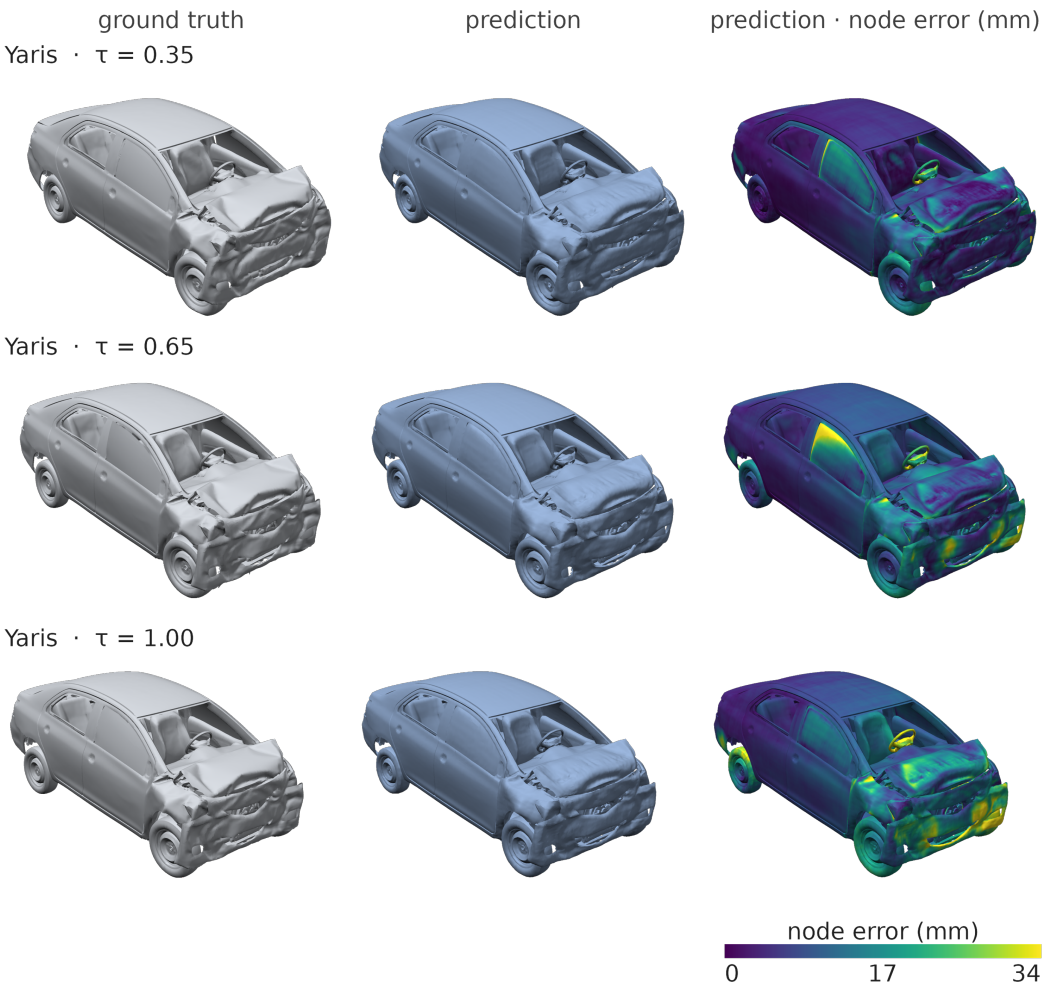

Figure 8: Full-field deformation prediction for a Yaris test simulation at three stages of the crash. ImpactFormer achieves an RMSE of 5.160 mm on this run. The node-wise error scale ranges from 0 to 34 mm.

Figure 9 shows the Silverado test run at three instants of the crash.

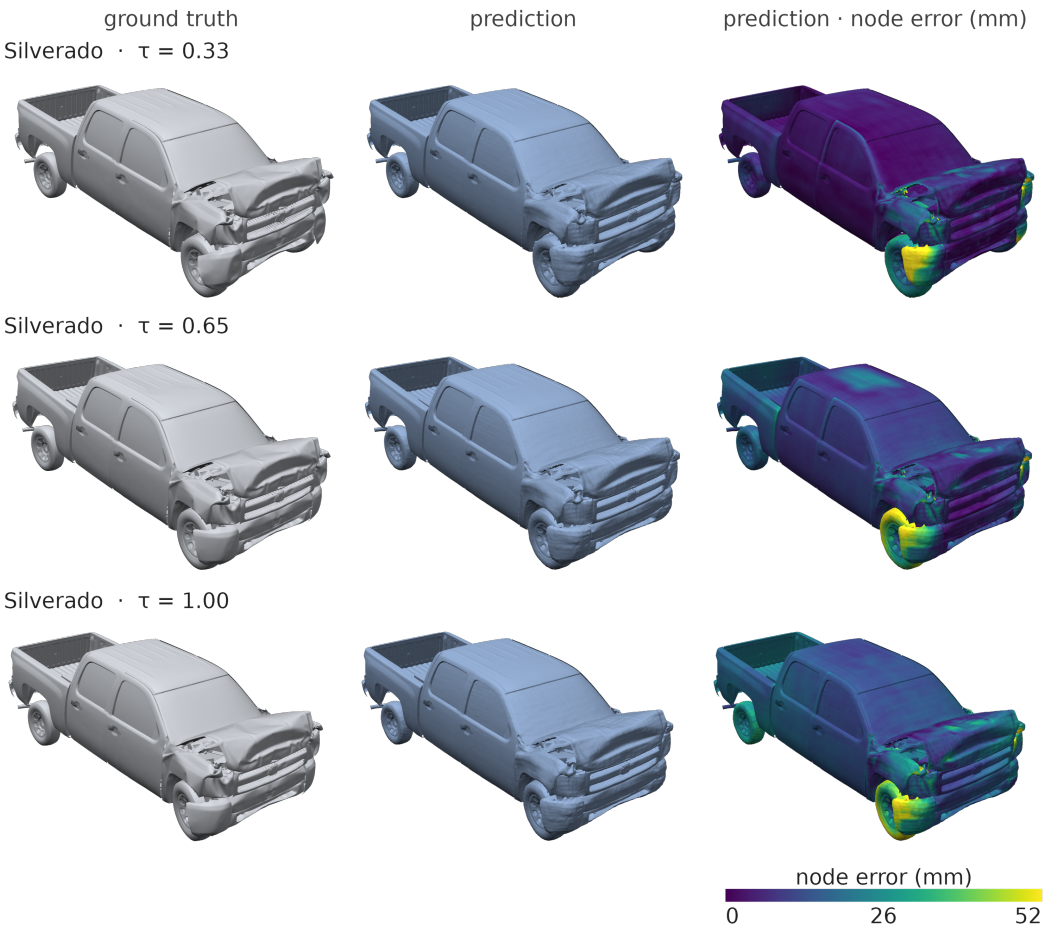

Figure 9: Full-field deformation prediction for a Silverado test simulation at three stages of the crash. ImpactFormer achieves an RMSE of 7.461 mm on this run. The node-wise error scale ranges from 0 to 52 mm.

Figure 10 shows the SHIFT-Crash test run at three instants of the crash.

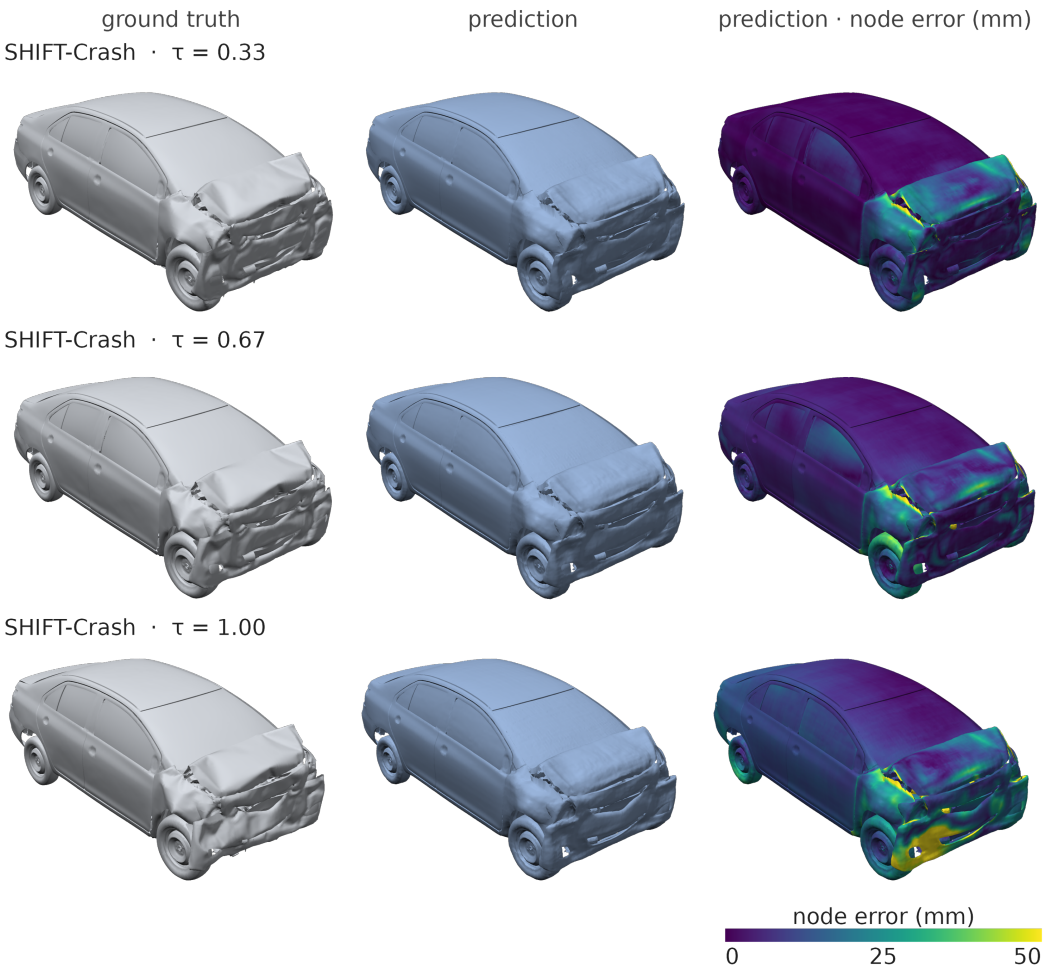

Figure 10: Full-field deformation prediction for a SHIFT-Crash test simulation at three stages of the crash. ImpactFormer achieves an RMSE of 6.650 mm on this run. The node-wise error scale ranges from 0 to 50 mm.