SCALING 3D PHYSICS SURROGATES: LOW-FIDELITY PRETRAINING FOR HIGH-FIDELITY TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Training neural surrogates for high-fidelity simulation requires expensive labels, limiting the scale of available supervision. We investigate low-fidelity simulation as a reusable source of pretraining data for unstructured 3D physics. Our approach is simple yet effective: using open-source fluid and solid-mechanics solvers, we generate approximate responses on source geometries constructed independently of downstream datasets, pretrain standard surrogate backbones, and finetune them on high-fidelity targets. Across unstructured downstream tasks spanning fluid fields, deformable-body responses, and physical systems whose governing equations are absent from pretraining, low-fidelity pretraining improves over training from scratch and competing pretraining baselines. Our pretraining can reduce high-fidelity data requirements by up to 77% at matched accuracy. Transfer remains effective when pretraining uses a single fixed physical condition and varies only geometry, demonstrating that this restricted source distribution already provides useful supervision for downstream adaptation.We further study scaling with pretraining-corpus size, model capacity, and physical-condition coverage, observing task-dependent improvements in downstream transfer. These results demonstrate low-fidelity simulation as an effective source of large-scale, reusable supervision for high-fidelity surrogates on unstructured 3D geometry.

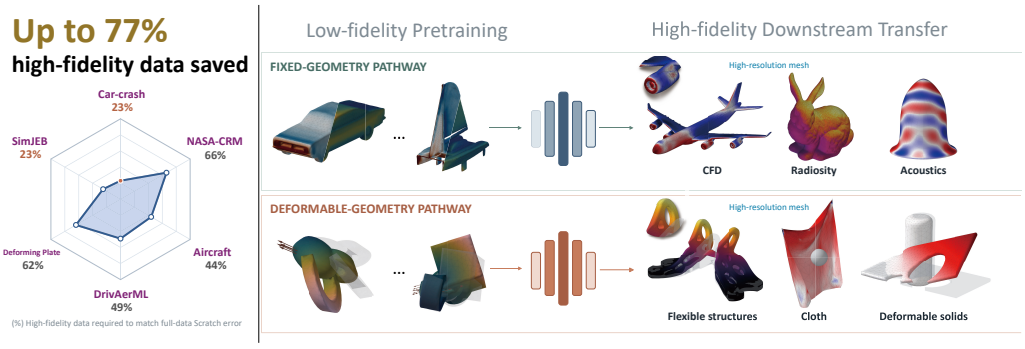


Figure 1: **Low-fidelity simulation pretraining for 3D physics surrogates.** A geometrically diverse corpus of inexpensive low-fidelity simulations is used to pretrain a transferable representation. The pretrained model benefits high-fidelity downstream tasks spanning steady flow, transient solid mechanics, and cross-physics prediction, while reducing the requirement of high-fidelity training data.

1 INTRODUCTION

Numerical simulation is central to engineering design, but high-fidelity solutions remain expensive. Predicting the flow around a vehicle or the deformation of a structure can take hours or days of CFD or finite-element computation. Neural surrogates amortize this cost by learning to predict physical responses from geometry and operating conditions (Wu et al., 2024; Pfaff et al., 2020). Training them, however, still requires substantial simulation data, so the cost of supervision becomes the

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.

bottleneck. In language and vision, pretraining on abundant, inexpensive data makes it possible to learn from smaller, curated target datasets (Kolesnikov et al., 2020; Raffel et al., 2020; Jia et al., 2021). This raises a central question for 3D physics: *what scalable source of supervision can reduce dependence on expensive high-fidelity simulations?*

Recent work explores several sources of pretraining supervision. On structured grids, models pretrained on simulated fields and trajectories transfer across PDEs and downstream data regimes (Herde et al., 2024; McCabe et al., 2023; Hao et al., 2024). Unstructured 3D problems introduce complex geometries that define the simulation domain and influence the spatial structure of the response. GeoPT (Wu et al., 2026) reports that static geometry-only pretraining with SDF or vector-distance targets provides little benefit or can cause negative transfer. It addresses this limitation by lifting geometric supervision with synthetic dynamics. These findings motivate pretraining targets that capture responses to conditions, beyond static shape reconstruction.

We take a direct route and use low-fidelity simulation as a scalable, transferable pretraining signal. Approximate solvers cheaply generate responses to geometry and boundary conditions at lower numerical fidelity than the target simulations. We generate fluid fields and solid-deformation responses over diverse geometries with open-source lattice-Boltzmann and finite-element solvers (Ataei & Salehipour, 2024; Meyer) respectively, pretrain standard surrogate backbones on their outputs, and finetune on high-fidelity tasks. The source corpora need not reproduce a particular downstream case and require no paired low- and high-fidelity simulations of downstream cases. This differs from learning a fidelity correction within a single application as the source checkpoint is reused across datasets with different geometries, output variables, and governing operators.

We evaluate this approach along three axes: task breadth, cross-physics transfer, and scaling. First, on unstructured 3D engineering benchmarks spanning aerodynamics and crash mechanics, CFD-pretrained surrogates improve over training from scratch and pretraining baseline. A separate solid-mechanics pretraining corpus extends the same principle to deformable-body prediction. Second, we finetune the CFD-pretrained model on radiosity, acoustic scattering, and electrostatics, whose governing operators differ from those used to generate the source data. Successful transfer to these tasks shows that low-fidelity pretraining can serve as a reusable initialization across physical systems. Finally, we study how transfer varies with pretraining-corpus size, model capacity, and downstream data availability. Low-fidelity pretraining reduces the high-fidelity supervision needed to reach a given accuracy. Expanding the source corpus and increasing model capacity yield further improvements on several tasks, with task-dependent scaling trends. Notably, a corpus generated under a single fixed physical condition already supports broad transfer. Adding more condition diversity can improve transfer further, depending on the downstream task.

Together, our contributions are threefold: **(1) Scalable low-fidelity pretraining**: we pretrain on large-scale fluid and solid responses, without paired downstream simulations. **(2) Data-efficient cross-physics transfer**: we improve downstream accuracy and reduce high-fidelity data requirements across physical systems. **(3) Empirical scaling and diversity analysis**: we study transfer as data scale and model capacity increase, and examine task-dependent effects of condition coverage.

2 Related Work

Transfer learning and scaling. Pretraining on a broad source distribution and adapting to a smaller target dataset is a standard paradigm in vision and language (Kolesnikov et al., 2020; Devlin et al., 2019; Raffel et al., 2020; He et al., 2022). Its benefit depends on the source distribution, pretraining scale, model capacity, and the amount of target supervision. Empirical scaling laws describe how performance changes with data, parameters, and compute (Kaplan et al., 2020; Hoffmann et al., 2022). Hernandez et al. (2021) quantify the benefit of pretraining as *effective data multiplier*. Meanwhile, pretraining can accelerates optimization while sufficiently trained scratch models will eventually close the gap (He et al., 2019).

Neural surrogates on general 3D geometries. Graph- and mesh-based simulators extend learned simulation to unstructured discretizations and time-dependent dynamics (Sanchez-Gonzalez et al., 2020; Pfaff et al., 2020). For large engineering systems, geometry-aware and attention-based operators enable prediction directly on complex 3D meshes (Hao et al., 2023; Li et al., 2023; Wu et al., 2024; Luo et al., 2025; Alkin et al., 2024; 2025; Wang et al., 2025), while recent datasets and

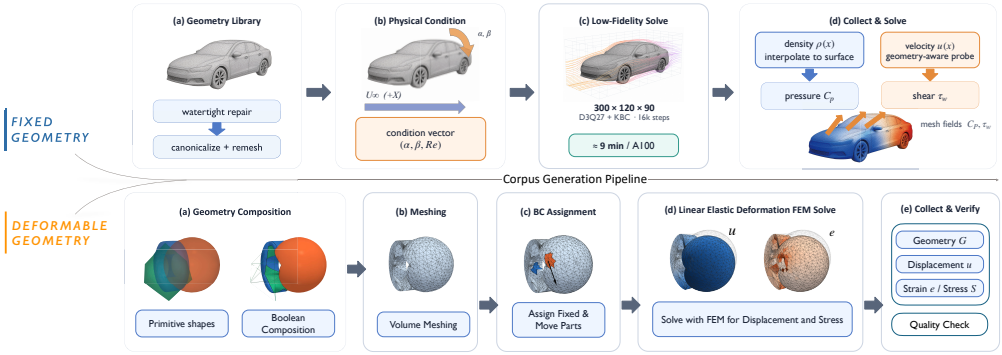


Figure 2: **Low-fidelity corpus generation.** Upper: Inexpensive XLB simulations convert geometries into mesh-aligned pressure and wall-shear-stress targets for fixed-geometry pretraining. Lower: Low-fidelity FEM simulations on geometries composed of primitive shapes for elastic deformation, used for deformable-body pretraining. Solved for displacement, strain and stress.

representations target industrial automotive aerodynamics (Elrefaie et al., 2024a; Chen et al., 2025; Elrefaie et al., 2024b). These works primarily advance geometric representation, architecture, and computational scale, for a particular type of downstream dataset. To complement these advances, we study how inexpensive physical supervision can provide reusable initialization across downstream tasks, while retaining standard backbones for field and sequence prediction.

Pretraining physical surrogates. Recent scientific foundation models pretrain neural operators on collections of equations or simulated trajectories (Subramanian et al., 2023; Herde et al., 2024; McCabe et al., 2023; Hao et al., 2024). They demonstrate transfer across physical systems and data regimes, but primarily in structured-grid settings. Pretrained weather and climate models similarly benefit from abundant gridded simulations and observations (Nguyen et al., 2023; Bodnar et al., 2024). Unstructured 3D problems introduce additional difficulty. Generic point-cloud pretraining learns shape representations (Yu et al., 2022; Pang et al., 2023), but geometry alone does not specify a system's response to forcing and boundary conditions. Recent work applies this idea directly to operator learning by reconstructing occupancy fields from unlabeled geometry and conditioning downstream operators on the learned latent representation (Zhang et al., 2026). GeoPT (Wu et al., 2026) augments solver-free geometric pretraining with synthetic dynamics. We take a different route: inexpensive numerical solvers directly provide dense physical fields or trajectories. This lets the same pretrain-and-finetune principle support standard backbones across Eulerian field prediction and Lagrangian deformable-body prediction.

Low-to-high-fidelity transfer. Multi-fidelity methods exploit correlations between inexpensive approximate simulations and costly accurate ones. Classical approaches combine fidelity levels through co-kriging (Kennedy & O'Hagan, 2000), nonlinear autoregressive corrections (Perdikaris et al., 2017), or physics-informed fusion (Meng & Karniadakis, 2020), as surveyed by Peherstorfer et al. (2018). Our work builds on this low-to-high-fidelity principle but studies low-fidelity simulation on unstructured 3D geometries as a broader source of pretraining supervision. The underlying numerical solvers are well established (Chen & Doolen, 1998; Ataei & Salehipour, 2024; Meyer).

3 Low-Fidelity Physics Pretraining

Overview. We pretrain standard surrogate backbones on inexpensive responses generated by low-fidelity solvers, then finetune them on high-fidelity downstream data. We instantiate this procedure in two settings: low-fidelity CFD surface fields provide the initialization for engineering and cross-operator field tasks, while low-fidelity solid-mechanics data support deformable-body prediction. The two settings use separate source corpora and pretrained checkpoints but follow the same pretrain-and-finetune procedure(Figure.2).

3.1 CONSTRUCTING THE LOW-FIDELITY CFD CORPUS

We construct a scalable source distribution by pairing ShapeNet vehicle geometries (Chang et al., 2015) with approximate surface responses,

$$(\mathcal{G}, \mathbf{c}) \longmapsto (C_p, \boldsymbol{\tau}_w)\,, \tag{1}$$

where $\mathcal{G}$ is a repaired, canonicalized, and remeshed surface and $\mathbf{c}$ specifies the flow condition. As illustrated in Figure 2, each geometry is simulated under K conditions with angle of attack $\alpha \in [-15^\circ, 15^\circ]$, sideslip $\beta \in [-20^\circ, 20^\circ]$, and Reynolds number in $[500, 5000]$. Body rotation realizes (α, β) relative to a fixed $+X$ freestream, which is supplied to the model as the condition input. Each case is voxelized on a $300 \times 120 \times 90$ Cartesian lattice and advanced for $16{,}000$ steps using XLB (Ataei & Salehipour, 2024), with a D3Q27 lattice and entropic KBC collision operator. We retain the density and velocity fields and map them back to the original surface. With $p = c_s^2 \rho$ and $\rho_\infty = 1$, the targets are

$$C_p(\mathbf{x}_s) = \frac{c_s^2(\rho(\mathbf{x}_s) - \rho_\infty)}{\frac{1}{2}\rho_\infty U_\infty^2}, \qquad \boldsymbol{\tau}_w(\mathbf{x}_s) \approx \mu \frac{(\mathbf{I} - \mathbf{n}\mathbf{n}^\mathsf{T})\mathbf{u}(\mathbf{x}_s + \delta \mathbf{n})}{\delta}, \tag{2}$$

where $\delta = 1.5\Delta x$. Using the original mesh normal avoids the grid-aligned normal error introduced by voxelized boundaries (Stahl et al., 2010; Matyka et al., 2013). We smooth the volumetric velocity with a Gaussian filter ($\sigma = 0.7\Delta x$) before interpolation and apply a bilateral filter to the reconstructed shear field. Appendix B provides more details about the corpus generation.

Corpus variants. We compare $K = 1, 4$, and 9 conditions per geometry. The $K = 1$ corpus uses one shared condition, while $K = 4$ and $K = 9$ sample the ranges above; the former is a subset of the latter. These variants support the geometry- and condition-coverage analyses in Section 6.

3.2 CONSTRUCTING THE LOW-FIDELITY DEFORMABLE CORPUS

We construct a scalable source distribution for 3D solid mechanics by pairing procedurally synthesized geometries with quasi-static elastic responses,

$$(\mathcal{G}, \mathbf{u}_{\mathrm{bc}}) \longmapsto (\mathbf{u}, \sigma_{\mathrm{vm}})\,, \tag{3}$$

where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a canonicalized volumetric tetrahedral mesh, $\mathbf{u}_{\mathrm{bc}}$ specifies surface Dirichlet boundary conditions, and $(\mathbf{u}, \sigma_{\mathrm{vm}})$ are nodal displacement and element-wise Von Mises stress fields.

As illustrated in Figure 2, each geometry is generated by composing 2–5 primitives (e.g., spheres, cylinders, boxes, tori) via randomized Constructive Solid Geometry (CSG) boolean union and difference operations. The resulting watertight surface is decimated to 1000–2000 vertices, normalized to $[-1, 1]^3$, and discretized into a 4-node tetrahedral volume mesh. To promote resolution and scale invariance, shapes undergo anisotropic scale jittering ($[0.1, 1.5]^3$) and random volume remeshing up to a $10\times$ node expansion factor. We then model linear isotropic elastostatics ($E = 1000\,\mathrm{Pa}, \nu = 0.3$) under infinitesimal strain. Boundary conditions are assigned to 2–4 disjoint surface patches grown via randomized BFS (10–50 nodes each), partitioned into *fixed regions* ($\mathbf{u} = \mathbf{0}$) and *moved regions* subjected to coupled rigid-body translations ($\leq 10\%$ bounding diameter) and rotations ($\leq 0.2\,\mathrm{rad}$). The discrete equilibrium system $\mathbf{K}\mathbf{u} = \mathbf{f}$ is solved across 10 ramp increments via GPU-accelerated sparse Conjugate Gradient in `torchfem` (Meyer), yielding the steady-state displacement field $\mathbf{u}$ and equivalent Von Mises stress σ_{vm}. Each simulation is structured as a heterogeneous graph with mesh edges and node-element incidence.

3.3 TRANSFER INTERFACE AND BACKBONE

We initialize each downstream model with all pretrained tensors whose names and shapes match the target architecture, while unmatched tensors remain randomly initialized. Our default pretraining interface follows the same input contract as the downstream models and GeoPT, allowing the input projection to transfer directly. Transolver (Wu et al., 2024) is used as the default backbone, with the GeoPT configuration of width 256, eight attention heads, an MLP ratio of 2, and 32 physics-aware slice tokens. Model-scaling experiments vary the depth over 8, 16, and 32 layers. Each backbone is pretrained separately, and checkpoints are transferred only within the same architecture.

3.4 GENERATION AND PRETRAINING COST AT SCALE

Table 1: **Downstream evaluation tasks.** The first two groups evaluate transfer from CFD pretraining, the third evaluates solid-mechanics pretraining. DrivAerML has a test set containing 50 samples.

Dataset	Physics / output	Condition $\mathbf{c}$	Train/eval.	Metric	Training pts
Engineering field-prediction benchmarks					
DrivAerML	Automotive aero.: C_p, $\boldsymbol{\tau}_w$	Fixed	400/34(50)	Rel-L^2	400 k
Aircraft	Aircraft aero.: C_p, p, $\mathbf{v}$, ρ	M_∞, α, β	100/50	Rel-L^2	30 k
NASA-CRM	Transonic aero.: C_p	M_∞, α	105/44	Rel-L^2	50 k
Car-Crash	Crash: $\max_t \sigma_{\mathrm{vm}}$	θ_{imp}	100/30	Rel-L^2	100 k
Cross-operator transfer tasks					
Radiosity	Light transport: radiosity B	$\hat{\boldsymbol{\ell}}$, ρ_d	480/120	Rel-L^2	100 k
Acoustics	Helmholtz: scattered pressure $\lvert p_{\mathrm{tot}}\rvert$	$\hat{\boldsymbol{k}}_{\mathrm{inc}}$, ka	400/100	Rel-L^2	20 k
Electrostatics	Electrostatics: surface charge	$\hat{\boldsymbol{E}}_\infty$	160/40	Rel-L^2	10 k
Extension to deformable-body mechanics					
DeformingPlate	Quasi-static deformation sequence	$\mathbf{q}_{\mathrm{act}}$	1000/100	Rollout RMSE	1 k
SphereSimple	Cloth dynamics trajectory	$\mathbf{q}_{\mathrm{sph}}$	1000/100	Rollout RMSE	2 k
SimJEB	Static displacement and stress	ℓ	756/764	RMSE	100 k

Both pipelines generate cases independently in parallel. A CFD case is one geometry–condition pair; a solid case contains ten ramp states on one geometry (20M states overall). The table reports generation time per case, including the published runtime of a typical DrivAerML simulation (Ashton et al., 2024). Pretraining times are measured task runtimes on A100 GPUs; A100-hours equal elapsed hours multiplied by GPU count. Source generation is the main upfront cost, while model pretraining takes hundreds of A100-hours. These reusable investments can be amortized across downstream applications.

Data generation (per case) Source	Scale	Time	Hardware
CFD	9.3–77k	542 s	1 A100
Solid	2M geom.	30 s	1 A100
DrivAerML	400	40 h	1,536 cores
Pretraining (per task) Model	Hours	A100-h	GPUs
CFD L8	50.0	200.0	4
CFD L16	72.2	288.9	4
CFD L32	41.0	327.8	8
AST 4M	102.8	822.2	8
AST 40M	100.0	800.0	8

4 Experimental Setup

We first evaluate CFD-pretrained surrogates on engineering field-prediction benchmarks and tasks governed by operators absent from the pretraining corpus. We then assess the broader applicability of the same pretraining principle through solid-mechanics pretraining and downstream deformation prediction. Table 1 summarizes these evaluation settings, including prediction targets, physical conditions, and dataset sizes. Detailed training and evaluation protocols are provided in Appendix C.

4.1 Downstream tasks

Engineering benchmarks. We evaluate CFD-pretrained models on four benchmarks. DrivAerML (Ashton et al., 2024) predicts surface pressure coefficient C_p and wall shear stress $\boldsymbol{\tau}_w$ under a fixed operating condition. Aircraft (Luo et al., 2025) predicts C_p, pressure p, velocity $\mathbf{v}$, and density ρ, conditioned on freestream Mach number M_∞, angle of attack α, and sideslip angle β. NASA-CRM (Bekemeyer et al., 2025) predicts C_p conditioned on M_∞ and α. Car-Crash (Wu et al., 2026) predicts the elementwise maximum von Mises stress over a collision, $\max_t \sigma_{\mathrm{vm}}$, conditioned on impact angle θ_{imp}.

Cross-operator tasks. We further evaluate on three tasks governed by operators absent from CFD pretraining. Radiosity predicts per-scene surface radiosity B from the non-emissive scene geometry, light direction $\hat{\boldsymbol{\ell}}$, and diffuse albedo ρ_d. Acoustics predicts scattered pressure $\lvert p_{\mathrm{tot}}\rvert$, conditioned on incident direction $\hat{\boldsymbol{k}}_{\mathrm{inc}}$ and nondimensional wavenumber ka. Electrostatics predicts induced surface charge, conditioned on the applied field direction $\hat{\boldsymbol{E}}_\infty$. Dataset details are provided in Appendix D.

Table 2: **Downstream transfer across model depths.** Relative L^2 error (%). Results are reported as mean $\pm$ std. Evaluation uses the test split for DrivAerML and the validation split for the other benchmarks. * denotes results quoted from the GeoPT paper.

	Method	DrivAerML		Aircraft	Car-Crash	NASA-CRM
		C_p	$\|\boldsymbol{\tau}_w\|$			
Scratch	Transolver L8	$4.37_{\pm 0.04}$	$7.40_{\pm 0.04}$	$5.08_{\pm 0.36}$	$18.36_{\pm 0.11}$	$9.20_{\pm 0.15}$
	Transolver L16	$3.90_{\pm 0.05}$	$6.68_{\pm 0.09}$	$4.98_{\pm 0.52}$	$17.36_{\pm 0.26}$	$9.36_{\pm 0.14}$
	Transolver L32	$3.66_{\pm 0.02}$	$6.28_{\pm 0.02}$	$5.31_{\pm 0.96}$	$18.95_{\pm 0.31}$	$9.93_{\pm 0.64}$
GeoPT	Transolver L8	$4.78_{\pm 0.02}$	$8.24_{\pm 0.04}$	$4.28_{\pm 0.11}$	$17.42_{\pm 0.03}$	$8.33_{\pm 0.12}$
	Transolver L16	–	–	8.64^*	16.61^*	8.68^*
	Transolver L32	–	–	8.36^*	16.57^*	8.51^*
Ours	Transolver L8	$4.16_{\pm 0.02}$	$7.12_{\pm 0.01}$	$4.16_{\pm 0.06}$	$15.98_{\pm 0.05}$	$7.97_{\pm 0.17}$
	Transolver L16	$3.85_{\pm 0.02}$	$6.53_{\pm 0.04}$	$4.11_{\pm 0.12}$	$14.73_{\pm 0.03}$	$7.84_{\pm 0.17}$
	Transolver L32	$3.72_{\pm 0.02}$	$6.32_{\pm 0.03}$	$4.09_{\pm 0.10}$	$13.88_{\pm 0.06}$	$8.60_{\pm 0.09}$

Deformable-body mechanics. We evaluate solid-mechanics pretraining on three benchmarks. DeformingPlate and SphereSimple (Pfaff et al., 2020) require autoregressively predicting quasi-static hyperelastic deformations (driven by actuators $\mathbf{q}_{\text{act}}$) and cloth dynamics (driven by spheres $\mathbf{q}_{\text{sph}}$), respectively. To simulate a challenging, data-constrained setting and improve training efficiency, we temporally subsample these sequences by factors of 10 and 2 (yielding 40 and 250 frames, respectively). Finally, SimJEB (Whalen et al., 2021) evaluates static displacement and stress prediction conditioned on load case ℓ.

4.2 Comparison protocols

Initialization comparisons. We compare three initialization strategies for Transolver (Wu et al., 2024): random initialization (Scratch), geometric pretraining (GeoPT), and low-fidelity CFD pretraining (Ours). For each task and model size, all three use the same downstream data and point-sampling pipeline. We evaluate GeoPT's released eight-layer checkpoint with the corrected implementation used for our downstream comparisons. The published L16/L32 values use the same reported settings, but their checkpoints have not been released, preventing re-evaluation with that correction. Aircraft, Car-Crash, NASA-CRM, and the cross-operator tasks use a common optimization schedule across initialization strategies. DrivAerML uses separate learning rates for scratch and pretrained models, with details in Appendix C. For deformable-body tasks, we compare randomly initialized and solid-pretrained AST models (Wang et al., 2025) at matching model sizes.

Full-data benchmark comparisons. On DrivAerML and NASA-CRM, we train with the complete training sets and evaluate predictions on the full surface meshes without point subsampling. We evaluate pressure and wall-shear-stress magnitude on DrivAerML, with pressure and skin-friction-coefficient magnitude on NASA-CRM. We compare these results with published task-specific methods following the corresponding benchmark evaluation conventions.

5 Task Evaluation

Transfer across engineering benchmarks As shown in Table 2, low-fidelity pretraining generally achieves lower errors than training from scratch and geometric pretraining across the benchmarks. In our repeated runs pretraining can also reduce run-to-run variability relative to scratch, particularly for the larger models. These results demonstrate improvements in both predictive accuracy and training consistency. On DrivAerML, the advantage diminishes at larger depth in the full-data regime, with Scratch slightly outperforming our L32 model. This is consistent with the observation that pretraining need not improve final accuracy when target supervision and optimization are sufficient (He et al., 2019). Its data-efficiency benefit is examined in Section 6.3. Figure 3 provides spatial comparisons under the controlled L8 protocol.

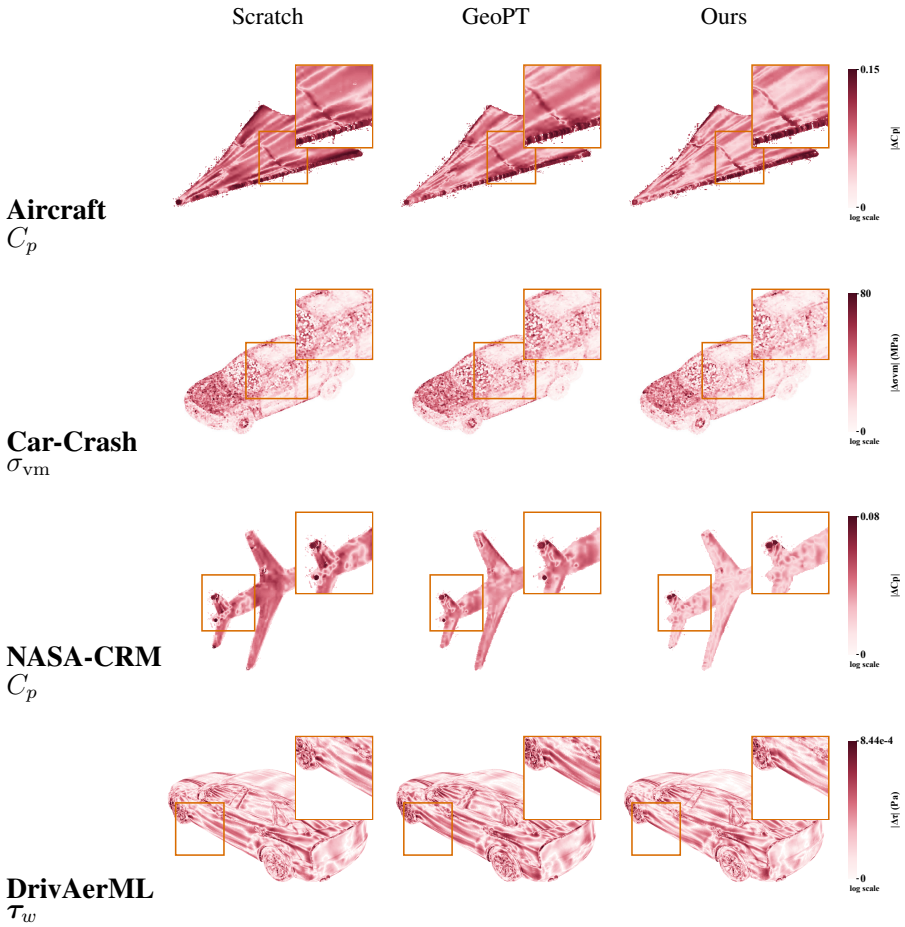


Figure 3: **Qualitative absolute errors.** Full prediction fields are provided in Appendix F.

Table 3: **Cross-operator transfer.** Validation relative L^2 error (%), reported as mean $\pm$ std.

Task	Scratch	GeoPT	Ours
Radiosity	23.71±0.62	21.77±0.10	**19.48±0.07**
Acoustics	22.49±0.42	22.76±0.43	**21.26±0.09**
Electrostatics	36.45±0.74	28.29±0.56	**25.70±0.15**
Radiosity control	10.91±0.04	10.84±0.03	10.80±0.02

Table 4: **Comparison with operator architectures.** Published baselines from Zhou et al. (2026).

	DrivAerML		NASA-CRM	
Method	C_p	$\|\tau_w\|$	C_p	C_f
AB-UPT (Alkin et al., 2025)	3.82	7.29	9.77	6.43
Transolver (Wu et al., 2024)	4.81	8.95	9.61	7.04
Transolver++ (Luo et al., 2025)	4.12	6.42	9.51	6.95
Transolver-3 (Zhou et al., 2026)	<u>3.71</u>	**5.85**	<u>8.71</u>	<u>5.85</u>
Ours, pretrained	**3.48**	<u>5.89</u>	**8.14**	**4.78**

Full-data comparison with task-specific methods When the full high-fidelity training sets are available, low-fidelity pretraining continues to improve downstream accuracy. As shown in Table 4, our pretrained model achieves the lowest errors on three of the four reported quantities, outperforming competitive task-specific architectures trained in the same full-data regime.

Transfer across governing operators As shown in Table 3, low-fidelity CFD pretraining outperforms both scratch and GeoPT on radiosity, acoustics, and electrostatics. These gains are obtained by pretraining directly on simulated fluid fields, without designing a specialized geometric pretext or modifying the pretraining objective for each downstream operator. Qualitative comparisons are provided in Appendix F.

Table 5: **Deformable tasks.** DeformingPlate and SphereSimple use trajectory-level rollout RMSE, while SimJEB uses RMSE for static displacement and stress.

	Method	DeformingPlate		SphereSimple	SimJEB	
		u	s		u	s
Scratch	AST 4M	$11.54_{\pm 0.28}$	$24755.35_{\pm 430.13}$	$46.29_{\pm 2.00}$	$1.64_{\pm 0.46}$	$101.93_{\pm 1.29}$
	AST 40M	$4.67_{\pm 3.19}$	$16289.05_{\pm 12608.21}$	$29.98_{\pm 0.67}$	$0.96_{\pm 0.44}$	$\mathbf{98.30_{\pm 2.65}}$
Ours	AST 4M	$4.69_{\pm 3.58}$	$13991.80_{\pm 7792.32}$	$43.41_{\pm 2.03}$	$0.54_{\pm 0.19}$	$98.40_{\pm 7.65}$
	AST 40M	$\mathbf{2.17_{\pm 0.05}}$	$\mathbf{6779.92_{\pm 157.56}}$	$\mathbf{28.59_{\pm 1.57}}$	$\mathbf{0.51_{\pm 0.09}}$	$100.72_{\pm 3.25}$

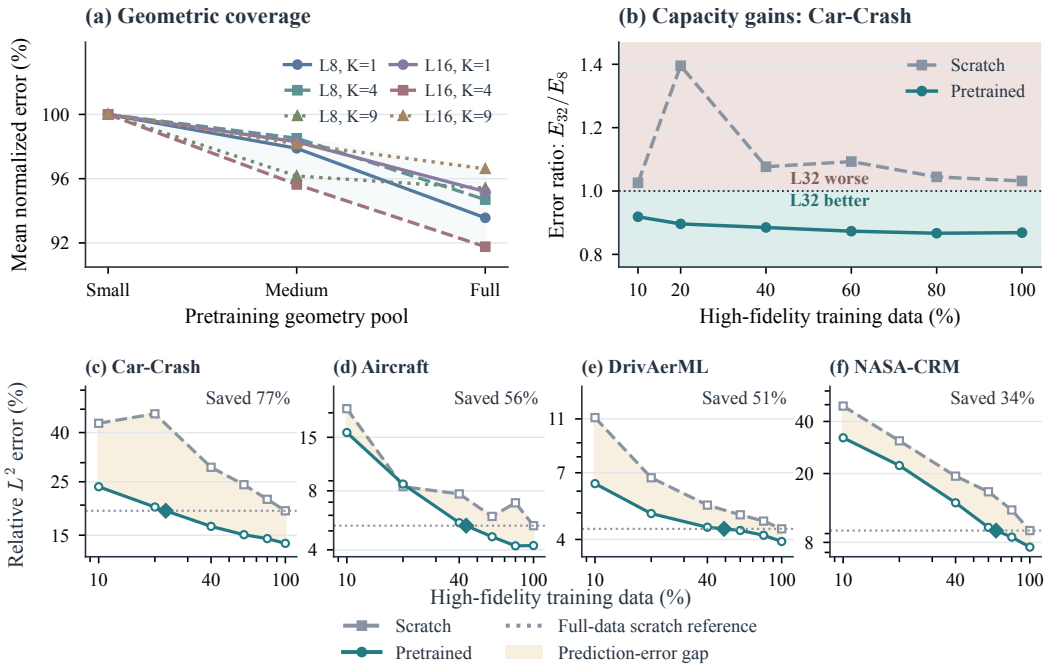


Figure 4: **Scaling and data efficiency of CFD pretraining.** **(a)** Six geometry sweeps, each averaging task-normalized errors over seven tasks. Shading spans the six curves. **(b)** Car-Crash L32/L8 error ratios; below one favors L32. **(c–f)** Best-available learning curves on log–log axes. Dotted lines mark full-data scratch accuracy; diamonds mark the interpolated pretrained crossings. Shading highlights prediction-error gaps at equal data budgets.

Extension to deformable-body mechanics Table 5 demonstrates our applicability by extending to direct field prediction and sequential deformation prediction. Solid-mechanics pretraining improves mean prediction accuracy across quasi-static deformation sequences, cloth dynamics, and static displacement prediction. Gains are less consistent for SimJEB stress prediction, indicating that transfer benefits depend on the target quantity.

6 Scaling Behavior and Transfer Analysis

We use CFD pretraining to examine how transfer depends on pretraining geometric coverage, model capacity, assess the resulting data efficiency and additional value of physical-condition diversity.

6.1 Scaling geometric coverage

We vary the number of pretraining geometries at each combination of model depth $L \in \{8, 16\}$ and conditions per geometry $K \in \{1, 4, 9\}$. Each of these six combinations defines a sweep over three geometry scales, with L and K held fixed within the sweep. As shown in Figure 4(a), broader

Table 6: **Physical-condition diversity in pretraining.** Relative L^2 error (%) for approximately equal-budget source allocations. Increasing K trades geometry for condition coverage.

		L8			L16		
Task	Fixed condition	$K=1$	$K=4$	$K=9$	$K=1$	$K=4$	$K=9$
DrivAerML (C_p)	✓	**4.53**	4.94	5.07	**4.18**	4.30	4.31
DrivAerML ($\|\tau_w\|$)	✓	**7.89**	8.63	8.87	**7.19**	7.38	7.37
Aircraft	×	4.20	4.00	**3.95**	4.00	**3.95**	4.02
NASA-CRM	×	8.23	**7.84**	8.01	8.36	**7.81**	8.03

coverage lowers downstream error in most comparisons. The gains recur across model depths and condition counts, supporting that expanding geometric diversity as a broadly effective way to improve transfer on the downstream tasks.

6.2 Model capacity and transfer

Low-fidelity pretraining enables larger-capacity models to better leverage their increased capacity, resulting in improved downstream transfer. With the source corpus held fixed, deeper pretrained models improve downstream on several tasks. The effect is particularly clear on Car-Crash (Figure 4(b)), where the L32/L8 error ratio remains below one for pretrained models but above one for scratch across every measured data fraction.

6.3 High-fidelity data efficiency

Figure 4(c–f) reports each task's best measured depth and pretraining configuration under a common matching rule. At the selected depth L_d, the reference is the error of Scratch trained on the complete target training set, $\epsilon_d^{\text{full}} = E_{\text{Scratch},L_d}(N_d)$. We estimate the pretrained fraction f_d needed to match this error by log–log interpolation between adjacent measured points, and define label saving as $S_d = 1 - f_d$. Under this definition, pretrained models save an estimated 34–77% of the full high-fidelity training sets. Three of the four tasks therefore match full-data Scratch accuracy with less than half of the original high-fidelity data.

6.4 The role of physical-condition diversity

The value of condition diversity depends on the variation that downstream models must resolve. DrivAerML varies geometry under a fixed operating condition, whereas NASA-CRM and Aircraft also require predictions conditioned on operating parameters such as Mach number and angle of attack. We therefore compare three approximately equal-budget allocations: more geometries with one shared condition ($K = 1$), an intermediate geometry pool with four conditions ($K = 4$), and fewer geometries with nine conditions ($K = 9$). The sampled conditions vary angle of attack, sideslip, and Reynolds number. Table 6 reports Relative L^2 error for these three variants. Fixed-condition pretraining already improves all reported targets, while additional conditions can provide further gains, while a consistent task-dependent pattern exists. DrivAerML favors geometry-heavy allocation, whereas NASA-CRM and Aircraft benefit from condition diversity.

7 Conclusion

Our work establishes inexpensive low-fidelity simulation as a scalable source of physically grounded supervision for neural surrogates on unstructured 3D geometry. Pretraining on approximate fluid and solid responses provides reusable initializations across downstream geometries, targets, and governing operators, while reducing high-fidelity data requirements. It also enables more effective use of model capacity on several tasks. Transfer benefits depend on source-corpus design, with geometric coverage and physical-condition diversity yielding task-dependent gains. These findings support large-scale, reusable simulation pretraining for data-efficient physics surrogates.

Limitations. Our current framework has not explicitly modeled volumetric flow structures. Extending the pretraining to incorporate volumetric physical fields is a direction for future work.

AI USE STATEMENT

Generative AI tools were used to improve the readability of the manuscript and to assist with LaTeX formatting. All AI-assisted content was reviewed and revised by the authors.

REPRODUCIBILITY STATEMENT

Dataset construction, preprocessing, hyperparameters, training, evaluation and additional implementation details are documented in the main paper and appendix. Code, dataset and pretrained checkpoints will be released upon publication.

ETHICS STATEMENT

This work uses engineering simulation data and does not involve human subjects or personally identifiable information. We are not aware of any direct ethical concerns arising from this work.

REFERENCES

Benedikt Alkin, Andreas Fürst, Simon Schmid, Lukas Gruber, Markus Holzleitner, and Johannes Brandstetter. Universal physics transformers: A framework for efficiently scaling neural operators. *Advances in Neural Information Processing Systems*, 37:25152–25194, 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. *arXiv preprint arXiv:2502.09692*, 2025.

Neil Ashton, Charles Mockett, Marian Fuchs, Louis Fliessbach, Hendrik Hetmann, Thilo Knacke, Norbert Schonwald, Vangelis Skaperdas, Grigoris Fotiadis, Astrid Walle, et al. Drivaerml: High-fidelity computational fluid dynamics dataset for road-car external aerodynamics. *arXiv preprint arXiv:2408.11969*, 2024.

Mohammadmehdi Ataei and Hesam Salehipour. Xlb: A differentiable massively parallel lattice boltzmann library in python. *Computer Physics Communications*, 300:109187, 2024.

Philipp Bekemeyer, Nathan Hariharan, Andrew M Wissink, and Jason Cornelius. Introduction of applied aerodynamics surrogate modeling benchmark cases. In *AIAA SCITECH 2025 Forum*, pp. 0036, 2025.

Cristian Bodnar, Wessel P Bruinsma, Ana Lucic, Megan Stanley, Johannes Brandstetter, Patrick Garvan, Maik Riechert, Jonathan Weyn, Haiyu Dong, Anna Vaughan, et al. Aurora: A foundation model of the atmosphere. *arXiv preprint arXiv:2405.13063*, 1(8), 2024.

AJ Burton and GF495032 Miller. The application of integral equation methods to the numerical solution of some exterior boundary-value problems. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 323(1553):201–210, 1971.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Qian Chen, Mohamed Elrefaie, Angela Dai, and Faez Ahmed. Tripnet: Learning large-scale high-fidelity 3d car aerodynamics with triplane networks. *arXiv preprint arXiv:2503.17400*, 2025.

Shiyi Chen and Gary D Doolen. Lattice boltzmann method for fluid flows. *Annual review of fluid mechanics*, 30(1):329–364, 1998.

Michael F Cohen, Shenchang Eric Chen, John R Wallace, and Donald P Greenberg. A progressive refinement approach to fast radiosity image generation. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pp. 75–84, 1988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Mohamed Elrefaie, Angela Dai, and Faez Ahmed. Drivaernet: A parametric car dataset for data-driven aerodynamic design and graph-based drag prediction. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 88360, pp. V03AT03A019. American Society of Mechanical Engineers, 2024a.

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. *Advances in Neural Information Processing Systems*, 37:499–536, 2024b.

Cindy M Goral, Kenneth E Torrance, Donald P Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *ACM SIGGRAPH computer graphics*, 18(3):213–222, 1984.

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 *International conference on machine learning*, pp. 12556–12569. PMLR, 2023.

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. *arXiv preprint arXiv:2403.03542*, 2024.

Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4918–4927, 2019.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel De Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *European conference on computer vision*, pp. 491–507. Springer, 2020.

Emmett Lalish. Manifold, 2026. URL `https://github.com/elalish/manifold`.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023.

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. *arXiv preprint arXiv:2502.02414*, 2025.

Maciej Matyka, Zbigniew Koza, and Łukasz Mirosław. Wall orientation and shear stress in the lattice boltzmann model. *Computers Fluids*, 73:115–123, 2013. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2012.12.018. URL https://www.sciencedirect.com/science/article/pii/S0045793013000078.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for physical surrogate models. *arXiv preprint arXiv:2310.02994*, 2023.

Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401:109020, 2020.

Nils Meyer. torch-fem: Gpu accelerated differentiable finite elements for solid mechanics with pytorch. URL https://github.com/meyer-nils/torch-fem.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.

Yatian Pang, Eng Hock Francis Tay, Li Yuan, and Zhenghua Chen. Masked autoencoders for 3d point cloud self-supervised learning. *World Scientific Annual Review of Artificial Intelligence*, 1: 2440001, 2023.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60(3):550–591, 2018.

Paris Perdikaris, Maziar Raissi, Andreas Damianou, Neil D Lawrence, and George Em Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings. Mathematical, physical, and engineering sciences*, 473(2198):20160751, 2017.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.

Hang Si. TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41(2):11:1–11:36, 2015. doi: 10.1145/2629697.

B. Stahl, B. Chopard, and J. Latt. Measurements of wall shear stress with the lattice boltzmann method and staircase approximation of boundaries. *Computers Fluids*, 39(9):1625–1633, 2010. ISSN 0045-7930. doi: https://doi.org/10.1016/j.compfluid.2010.05.015. URL https://www.sciencedirect.com/science/article/pii/S0045793010001283.

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. *Advances in Neural Information Processing Systems*, 36:71242–71262, 2023.

Hao Wang, Yu Liu, Daniel Biggs, Haoru Wang, Jiandong Yu, and Ping Huang. Learning deformable body interactions with adaptive spatial tokenization. *arXiv preprint arXiv:2507.13707*, 2025.

Eamon Whalen, Azariah Beyene, and Caitlin Mueller. Simjeb: simulated jet engine bracket dataset. In *Computer Graphics Forum*, volume 40, pp. 9–17. Wiley Online Library, 2021.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

Haixu Wu, Minghao Guo, Zongyi Li, Zhiyang Dou, Mingsheng Long, Kaiming He, and Wojciech Matusik. Geopt: Scaling physics simulation via lifted geometric pre-training. *arXiv preprint arXiv:2602.20399*, 2026.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19313–19322, 2022.

Zhizhou Zhang, Youjia Wu, Kaixuan Zhang, and Yanjia Wang. From cheap geometry to expensive physics: A physics-agnostic pretraining framework for neural operators. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

APPENDIX

A OVERVIEW

The appendix provides details supporting the main results. Appendix B describes source-corpus construction, numerical solvers, preprocessing, and generation costs. Appendix C documents model architectures, parameter transfer, and training and evaluation protocols. Appendix D details the cross-operator datasets, condition inputs, target transformations, and solver validation. Appendix E reports additional scaling results and the curves underlying the data-saving estimates. Appendix F presents qualitative comparisons.

B CORPUS GENERATION DETAILS

This section documents the construction of the source data used for pretraining. The CFD and solid-mechanics experiments use independent corpora and checkpoints. We first describe the CFD corpus, then document the solid-mechanics corpus.

B.1 CFD SOURCE CORPUS

Source geometries and preprocessing. The source geometry collection is drawn from multiple vehicle-related categories in ShapeNet (Chang et al., 2015), which is not restricted to car meshes. Each input mesh is repaired to obtain a closed surface, transformed to a common wind-tunnel coordinate frame, and remeshed before simulation. The canonical frame uses the first coordinate as the streamwise direction and the third coordinate as the vertical direction. Geometry identity is preserved throughout corpus construction so that training and validation can be split by mesh rather than by simulation case. Consequently, different physical conditions applied to one geometry never appear on opposite sides of a pretraining split. The original source collection contains approximately 11,000 candidate geometries. All the experiments operate on retained, geometry-disjoint indices after preprocessing and simulation filtering, containing 9,336 training and 1,167 validation geometries.

Physical-condition sampling. Each simulation pairs a geometry $\mathcal{G}$ with a flow condition $\mathbf{c}$. We keep the freestream direction fixed in the lattice frame and rotate the body to realize angle of attack $\alpha \in [-15^\circ, 15^\circ]$ and sideslip $\beta \in [-20^\circ, 20^\circ]$. The simulation Reynolds number is sampled in $[500, 5000]$. After simulation, geometry, normals, and vector-valued targets are returned to the canonical body frame. The model condition contains the corresponding freestream direction and a normalized speed scalar, broadcast to every surface point. We construct three variants with $K \in \{1, 4, 9\}$ conditions per geometry. For $K = 1$, every geometry uses one shared condition, isolating the supervision obtained from geometric variation. The $K = 4$ and $K = 9$ variants sample the ranges above, and the four-condition cases are retained as a subset of the nine-condition corpus.

Low-fidelity flow solver. We generate approximate flow responses with the GPU-accelerated XLB implementation of the lattice-Boltzmann method (Ataei & Salehipour, 2024). The generation environment uses XLB 0.2.1 and Warp 1.7.0. Each transformed body is voxelized on a $300 \times 120 \times 90$ Cartesian lattice. Simulations use a D3Q27 velocity stencil and the entropic KBC collision operator, which provides robustness at the deliberately coarse resolutions used for corpus generation. Each case is advanced for 16,000 lattice steps, after which the density $\rho(\mathbf{x})$ and velocity $\mathbf{u}(\mathbf{x})$ are retained for surface reconstruction. A case takes approximately 10 minutes on one NVIDIA A100 under this configuration.

Surface-field reconstruction. XLB evolves density and velocity on the Cartesian lattice, whereas pretraining supervises fields on the original surface mesh. We recover pressure from the lattice equation of state, $p = c_s^2 \rho$. With $\rho_\infty = 1$, the pressure coefficient at surface point $\mathbf{x}_s$ is

$$C_p(\mathbf{x}_s) = \frac{c_s^2\big(\rho(\mathbf{x}_s) - \rho_\infty\big)}{\frac{1}{2}\rho_\infty U_\infty^2}, \tag{4}$$

where the surface density is interpolated from the surrounding lattice values.

Wall shear stress requires a near-wall velocity gradient and is more sensitive to voxelization. A curved body becomes a staircase boundary on the lattice, so we use the normal $\mathbf{n}$ of the original mesh rather than a grid-aligned boundary normal. We sample at $\mathbf{x}_p = \mathbf{x}_s + \delta\mathbf{n}$ with $\delta = 1.5\Delta x$, project the sampled velocity onto the tangent plane, and use the no-slip wall value to form a one-sided gradient:

$$\mathbf{u}_t(\mathbf{x}_p) = \left(\mathbf{I} - \mathbf{n}\mathbf{n}^{\mathsf{T}}\right)\mathbf{u}(\mathbf{x}_p), \tag{5}$$

$$\boldsymbol{\tau}_w(\mathbf{x}_s) \approx \mu\frac{\mathbf{u}_t(\mathbf{x}_p)}{\delta}. \tag{6}$$

Before interpolation, the volumetric velocity is smoothed with a Gaussian filter of standard deviation $0.7\Delta x$. An edge-preserving bilateral filter is then applied on the reconstructed surface field. These operations suppress lattice-aligned oscillations amplified by differentiation while preserving larger spatial variations. The same reconstruction parameters are used for every geometry and physical condition.

Stored representation and pretraining interface. The stored labels comprise one pressure channel and three wall-shear-stress components. Training normalizes each target channel using statistics computed from the training split. Each training case is sampled to 32,768 surface points before batching; validation operates on the complete surface through a tiled predictor. Positions are normalized per geometry by scaling the streamwise extent to five, centering the streamwise coordinate by its mean and the lateral coordinate by its bounding-box midpoint, and shifting the minimum vertical coordinate to zero.

The pointwise model input follows the 11-dimensional transfer interface used in the downstream experiments,

$$[\,\mathbf{x}\ (3),\ 0\text{-pad}\ (1),\ \mathbf{n}\ (3),\ \mathbf{v}\ (4)\,], \tag{7}$$

where $\mathbf{v}$ contains the freestream direction and normalized speed. Before packing this interface, the first two position, normal, and shear components are sign-flipped to reconcile the solver frame with the downstream body-motion convention. This fixed transformation is applied consistently to training samples, validation samples, and normalization statistics.

Quality control and scope. Geometry-disjoint validation prevents different conditions of the same mesh from inflating pretraining validation performance. Corpus generation also checks that the initial transient has left the computational domain before the retained snapshot. The final-snapshot choice was assessed using the variation of integrated drag and reconstructed wall shear stress over a late-time window; time averaging was not applied to the released targets. We additionally inspected the sensitivity of wall-shear reconstruction to the probe distance and smoothing strength before fixing the shared parameters above. These checks are intended to reject unstable labels and select a consistent reconstruction recipe. They do not turn the coarse LBM solutions into high-fidelity CFD: the source fields remain approximate supervision whose value is assessed through downstream transfer.

Generation throughput and aggregate cost. The complete 16,000-step XLB solve takes approximately 542 s per case on one NVIDIA A100, or about nine minutes for one geometry–condition pair. At this rate, the retained training portions of the $K = 1$, $K = 4$, and $K = 9$ corpora correspond to approximately 1.41, 5.16, and 11.60 thousand A100-hours of simulation, respectively. Including the 1,167 held-out geometries increases the fixed-condition corpus total from 1.41 to 1.58 thousand A100-hours. These values are aggregate accelerator times obtained by summing independent retained cases, rather than elapsed wall-clock time. Corpus generation is parallel over geometry–condition pairs, so elapsed time decreases with the number of available workers. The estimates also exclude rejected candidates and the subsequent cost of model pretraining.

B.2 Solid-mechanics source corpus

Procedural geometry synthesis and remeshing. Mirroring the role of vehicle meshes in the CFD corpus (Section B.1), the solid-mechanics source distribution requires high topological diversity without manual CAD authoring. We procedurally synthesize geometries using Constructive Solid Geometry (CSG) executed with Manifold3D (Lalish, 2026). Each body is formed by sampling

$N_p \in [2, 5]$ basic primitives chosen uniformly from eight canonical geometric classes: spheres, cylinders, boxes, tori, cones, capsules, annuli, and icosahedra. Each primitive is assigned random spatial dimensions, positioned within $[0.3, 0.7]^3$, rotated by an arbitrary angle about a random unit axis, and composed via sequential boolean unions and differences. The resulting watertight 2-manifold surface is extracted, decimated to 1000–2000 boundary vertices while preserving topology, and checked for manifold validity. We then discretize the enclosed domain into a 4-node tetrahedral volume mesh using TetGen (Si, 2015) with Delaunay quality constraints (radius-edge ratio ≤ 1.5, minimum dihedral angle $\geq 5^\circ$). To prevent neural surrogates from overfitting to fixed coordinate scales or discretization densities, each volume mesh undergoes data augmentation: vertices are centered and canonicalized to $[-1, 1]^3$, subjected to anisotropic scaling in $[0.1, 1.5]^3$, and stochastically remeshed to target an up to $10\times$ node expansion factor via element volume bounding.

Boundary condition and deformation sampling. In contrast to the external freestream conditions $\mathbf{c}$ governing the CFD simulations, solid-mechanics cases are driven by heterogeneous displacement boundary conditions $\mathbf{u}_{\text{bc}}$ applied directly to the surface manifold. For each mesh, boundary surface nodes are identified from the outer tetrahedral facets. We extract $K_g \in [2, 4]$ mutually disjoint node patches by initiating breadth-first searches (BFS) from randomly chosen surface seeds, sampling patch sizes between 10 and 50 nodes along the mesh adjacency graph. Candidate patches are partitioned into fixed anchor regions $\mathcal{V}_{\text{fixed}}$ and actuated regions $\mathcal{V}_{\text{moved}}$. Fixed nodes enforce a homogeneous Dirichlet condition $\mathbf{u}(\mathbf{x}) = \mathbf{0}$. For each actuated patch, we prescribe a coupled rigid-body motion parameterized by a translation $\mathbf{t}$ and rotation $\mathbf{R}$:

$$\mathbf{u}_{\text{bc}}(\mathbf{x}) = \mathbf{R}\,(\mathbf{x} - \bar{\mathbf{x}}) + \mathbf{t} - (\mathbf{x} - \bar{\mathbf{x}})\,, \qquad \forall \mathbf{x} \in \mathcal{V}_{\text{moved}}, \tag{8}$$

where $\bar{\mathbf{x}} = \frac{1}{|\mathcal{V}_{\text{moved}}|} \sum_{\mathbf{x}_i \in \mathcal{V}_{\text{moved}}} \mathbf{x}_i$ is the patch centroid. The translation $\mathbf{t}$ is drawn uniformly with magnitude up to 10% of the mesh bounding diameter, while $\mathbf{R}$ represents a rotation about a random unit axis with angle $\theta \in [-0.2, 0.2]$ rad. All unconstrained surface and interior nodes remain free ($\mathbf{u} =$ unconstrained), ensuring diverse bending, twisting, and compressive deformation modes.

Low-fidelity solid mechanics solver and loading schedule. We simulate static deformation under linear isotropic elasticity and infinitesimal strain kinematics, governed by the Navier-Cauchy equilibrium equations:

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}, \qquad \boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} = \tfrac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathsf{T}}\right), \tag{9}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\boldsymbol{\varepsilon}$ is the infinitesimal strain tensor, and $\mathbf{C}$ is the fourth-order isotropic stiffness tensor parameterized by Young's modulus $E = 1000\,\text{Pa}$ and Poisson's ratio $\nu = 0.3$. The discretized global equilibrium system $\mathbf{K}\mathbf{u} = \mathbf{f}$ is solved in double precision using GPU-accelerated sparse Conjugate Gradient implemented in `torchfem`. Analogous to tracking time-dependent convergence in CFD, we apply boundary displacements incrementally across $S = 10$ linear quasi-static ramp load steps:

$$\mathbf{u}_{\text{bc}}^{(s)} = \frac{s}{S}\mathbf{u}_{\text{bc}}, \qquad s \in \{1, 2, \ldots, S\}. \tag{10}$$

At each step s, the linear system is solved to convergence, and we evaluate the nodal displacement field $\mathbf{u}^{(s)} \in \mathbb{R}^{N\times 3}$ alongside element-wise Cauchy stress tensors $\boldsymbol{\sigma}^{(s)} \in \mathbb{R}^{M\times 3\times 3}$. From the stress tensor, we recover the scalar Von Mises equivalent stress:

$$\sigma_{\text{vm}}^{(s)} = \sqrt{\tfrac{1}{2}\left[(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2 + 6(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{31}^2)\right]}. \tag{11}$$

Generation throughput and aggregate cost. The full data generation pipeline consists of two asynchronous stages: shape generation and elastic deformation simulation. Shape generation runs entirely on the CPU and is computationally inexpensive, taking approximately 0.6–1.5 s per shape depending on process concurrency. With 16 parallel workers, the shape generation throughput is approximately 11 shapes/s. The simulation stage takes approximately 30 s per mesh on an NVIDIA A100 GPU. The runtime is dominated by the multi-step linear FEM solve, which takes 26 s on average (87% of the total runtime) and varies with mesh resolution, ranging from 5.6 to 64 s across configurations. The remaining stages incur relatively small overheads: surface/volume remeshing (≈ 0.6 s), solid model construction (≈ 0.4 s), boundary-condition sampling (≈ 0.25 s), and graph serialization and I/O (≈ 2.5 s). Applying this protocol to 2,000,000 distinct procedural geometries

produces $2\text{M} \times 10 = 20\text{M}$ pretraining equilibrium states. At an average simulation cost of 30 s per mesh, generating the full dataset requires approximately 16,667 A100 GPU-hours, or 694 A100 GPU-days. We use 128 A100 GPUs for data generation, leading to approximately 5.4 days of GPU wall-clock time.

Graph representation, AST interface, and quality control. Simulations are exported into heterogeneous graphs (`HeteroData`) compatible with our AST backbone. Nodes represent tetrahedral mesh vertices $\mathbf{x} \in \mathbb{R}^3$, carrying a categorical boundary flag $c \in \{0, 1, 2\}$ (unconstrained, fixed, or actuated), the prescribed displacement $\mathbf{u}_{\text{bc}} \in \mathbb{R}^3$, and ground-truth target displacement $\mathbf{u}^{(s)} \in \mathbb{R}^3$. Graph edges encompass both undirected spatial mesh edges connecting adjacent vertices in the tetrahedral decomposition, and bipartite incidence edges linking elements to their four constituent nodes. Element nodes store the target scalar Von Mises stress $\sigma_{\text{vm}}^{(s)} \in \mathbb{R}$. Both single-step targets and complete multi-step trajectory sequences are retained. Rigorous quality control is enforced at every stage: cases encountering non-manifold CSG intersections, inverted tetrahedra with non-positive scaled Jacobians, TetGen constraint recovery failures, or Conjugate Gradient divergence are automatically discarded and resampled with replacement, guaranteeing that all retained samples represent physically valid, stable elastostatic equilibria.

C Experimental details

This section specifies the model interfaces, task-specific training and evaluation protocols, and the construction of the controlled scaling experiments, which are used in theexperiments reported in Sections 4 to 6.

C.1 Model architectures and parameter transfer

Field-prediction models. The engineering and cross-operator experiments use Transolver (Wu et al., 2024). The main comparisons use L8, L16, and L32 variants with 8, 16, and 32 transformer blocks, respectively. All have hidden width 256, eight attention heads, MLP ratio 2, and 32 physics-attention slice tokens. The controlled common-corpus capacity study additionally includes an L4 model. Network depth is the only architectural variable within that study.

The pointwise input follows the 11-dimensional contract used during CFD pretraining,

$$[\,\mathbf{x}\ (3),\ \text{0-pad}\ (1),\ \mathbf{n}\ (3),\ \mathbf{v}\ (4)\,], \tag{12}$$

where $\mathbf{x}$ is the normalized position, $\mathbf{n}$ is the outward normal, and $\mathbf{v}$ broadcasts the case-level physical condition to all surface points. Positions also enter the spatial pathway of the first Transolver block. Task-specific definitions of $\mathbf{v}$ are summarized in Table 1; the cross-operator mappings are detailed in Appendix D.

Deformable-body models. The deformable experiments use the 4M- and 40M-parameter variants of Adaptive Spatial Tokenization (AST) (Wang et al., 2025). At each step, the model consumes the current normalized mesh state and node type. It predicts a displacement increment, together with stress when the target task contains stress labels. DeformingPlate and SphereSimple apply this update autoregressively, whereas SimJEB uses the same representation for a single-state structural response.

Transferred parameters. For both model families, finetuning loads every source tensor whose parameter name and shape match the target model. The compatible input projection and backbone parameters are therefore transferred, while task-specific output layers are reinitialized when the target dimensionality changes. Scratch and pretrained models otherwise share the downstream architecture. The electrostatics experiment has one documented exception in an unused constant condition slot, see Appendix D.3.

C.2 Task-specific evaluation protocols

Common field-prediction protocol. Aircraft, Car-Crash, NASA-CRM, radiosity, acoustics, and electrostatics use the same optimization schedule across initialization strategies. Models are trained

for 200 epochs with AdamW, a one-cycle learning-rate schedule with a 30% warm-up fraction, weight decay 10^{-5}, and batch size 1. Surface points are randomly sampled at each training step according to Table 7. The checkpoint with the lowest validation relative-L^2 error is retained.

Table 7: **Task-specific Transolver finetuning settings.** The 50,000-point cap is active for radiosity and inactive for the smaller native acoustics and electrostatics meshes.

Task	Peak LR	Points per case	Outputs
Aircraft	4×10^{-4}	30,000	6
Car-Crash	1×10^{-3}	100,000	1
NASA-CRM	1×10^{-3}	50,000	1
Radiosity	1×10^{-3}	50,000	1
Acoustics	1×10^{-3}	full mesh ($\sim 20,000$)	1
Electrostatics	1×10^{-3}	full mesh ($\sim 10,000$)	1

Field metrics. For a case i, the reported relative error is

$$e_i = \frac{\|\widehat{\mathbf{y}}_i - \mathbf{y}_i\|_2}{\|\mathbf{y}_i\|_2}, \qquad E = \frac{1}{N}\sum_{i=1}^{N} e_i. \tag{13}$$

The norm includes all evaluated surface points and output channels for the reported quantity. Engineering targets are evaluated after inversion of their training normalization. Radiosity is evaluated in its per-scene standardized space; electrostatics is evaluated after the signed-log and per-scene standardization described in Appendix D.3; acoustics is evaluated on the de-normalized total-pressure magnitude $|p_{\mathrm{tot}}|$.

DrivAerML. DrivAerML uses 400 training, 34 validation, and 50 test cases. At each training step, 400,000 surface points are sampled per case. The global batch size is 4 across four GPUs. Positions are normalized by the geometry bounding box, and each target channel is standardized using statistics from the training set. Optimization uses AdamW with weight decay 0.05, a cosine schedule with 5% linear warm-up, and gradient-norm clipping at 1.0.

The main initialization comparison trains for 200 epochs. Scratch uses peak learning rate 10^{-3}, while GeoPT and our pretrained initialization use 5×10^{-4}. The full-data architecture comparison trains L32 for 500 epochs and uses peak learning rate 10^{-3} for both scratch and pretrained models. The data-efficiency experiment also uses the learning-rate-matched protocol. Evaluation processes every test surface without subsampling using 400,000-point tiles. Pressure is scored directly, and wall shear stress is scored through its direction-invariant magnitude $\|\boldsymbol{\tau}_w\|$ (Zhou et al., 2026).

NASA-CRM full-mesh evaluation. The architecture comparison evaluates all 44 cases on the complete 454,000-cell surface mesh. Surface pressure C_p and skin-friction coefficient magnitude $\|C_f\|$ are reported separately at the checkpoint selected by validation performance.

Cross-operator tasks. Radiosity, acoustics, and electrostatics use the common 200-epoch protocol in Table 7. Their numerical solvers, geometry-disjoint splits, condition encodings, target transformations, and validation tests are given in Appendix D.

Deformable-body tasks. DeformingPlate and SphereSimple use 1,000 training, 100 validation, and 100 test trajectories. Training minimizes one-step prediction error, and Gaussian noise with standard deviation 0.003 is added to the input displacement on non-boundary nodes. DeformingPlate predicts displacement increments and von Mises stress with normalized MSE weights 1.0 and 0.01. SphereSimple predicts the displacement increment with unit weight. For efficiency, we subsampled the DeformingPlate and SphereSimple sequences by factors of 10 and 2, respectively, reducing the original 400 and 500 frames to 40 and 250 frames per sequence. We report trajectory RMSE after aggregating over rollout steps and mesh nodes. Both AST(Wang et al., 2025) sizes are trained for 200 epochs with batch size 32 on the two trajectory tasks. Their configured base learning rate is 2×10^{-5} with square-root batch-size scaling, followed by 1,000 warm-up steps and a cosine schedule with a 100-epoch restart period, here weight decay is zero.

SimJEB is evaluated as a static prediction problem under vertical, horizontal, diagonal, and torsional load cases. The reported split contains 756 training and 764 evaluation geometry–load pairs, corresponding to 189 and 191 retained geometries with four load cases each. The model predicts the three displacement components and von Mises stress, with normalized MSE weights 1.0 and 0.01. SimJEB models are trained for 200 epochs with batch size 4, configured base learning rate 10^{-5} with square-root scaling, weight decay 10^{-5}, 1,000 warm-up steps, and a cosine schedule with a 100-epoch restart period. Static displacement and stress are reported as RMSE over the evaluation set.

C.3 Scaling and data-efficiency protocols

Source-corpus grid. The controlled CFD experiments in 4(a) combine model depth $L \in \{8, 16\}$ with conditions per geometry $K \in \{1, 4, 9\}$. Within each (L, K) sweep, geometric coverage uses nested small, medium, and full pools. The small and medium pools contain 1,091 and 2,456 training geometries. The full fixed-condition pool contains 9,336 geometries. Validation is always disjoint by geometry. All grid pretraining runs use AdamW with peak learning rate 2×10^{-3}, global batch size 16 across four GPUs, 2,500 warm-up steps, a cosine schedule, and fp32 computation. The scheduled budgets in thousands of steps for small/medium/full are 120/200/600 for $K = 1$, 180/600/700 for $K = 4$, and 600/400/1,000 for $K = 9$, at both depths. Consequently, the geometry curves characterize the tested corpus-expansion recipes rather than an iso-compute scaling law. The approximately equal-budget allocation comparison separately contrasts full-$K = 1$, medium-$K = 4$, and small-$K = 9$ configurations at 600,000 scheduled steps.

Capacity comparison. The controlled capacity experiment uses the same full-geometry, $K = 1$ training index and approximately 600,000 scheduled pretraining steps for L4, L8, L16, and L32 models. Width, input processing, downstream architecture, and downstream training protocol remain fixed; only depth changes.

High-fidelity data fractions. For each engineering benchmark, we construct nested training subsets at 10%, 20%, 40%, 60%, and 80% of the complete training set; the 100% point is the corresponding full-data result. A single fixed permutation is used and every larger subset contains all examples from the preceding fractions. Validation and test sets remain unchanged. Fractions count simulation cases; for Aircraft, a case is a geometry–condition pair. DrivAerML rescales warm-up duration to preserve the 5% fraction of total optimization steps.

The main figure and teaser use the best reported model depth and source-checkpoint configuration for each task. Each selected pretrained curve is matched to full-data Scratch error at that same depth. For adjacent fractions f_0, f_1 with pretrained errors e_0, e_1 bracketing the Scratch target e_*,

$$\log f_* = \log f_0 + \frac{\log e_* - \log e_0}{\log e_1 - \log e_0}(\log f_1 - \log f_0), \qquad S = 1 - f_*. \tag{14}$$

Only measured intervals are interpolated. Curve points average the available runs of the same listed configuration.

Physical-condition diversity. The condition comparison evaluates $K = 1, 4, 9$ source checkpoints under unchanged downstream protocols. The $K = 1$ corpus uses a common fixed flow condition; $K = 4, 9$ vary angle of attack, sideslip, and Reynolds number.

D Cross-operator datasets and solver validation

The cross-operator experiments test whether a CFD-pretrained initialization remains useful after supervised finetuning on operators absent from the source corpus. All three datasets use unstructured surface meshes and the same 11-dimensional point interface as the CFD experiments. The task-specific condition occupies the four condition slots, and the scalar target replaces the CFD output head. Table 8 summarizes the final corpora used in Table 3. The ShapeNet geometries used here come from furniture and object categories that are disjoint by category and model identity from the vehicle categories used for CFD pretraining.

Table 8: **Cross-operator corpus summary.** Counts and normalization correspond to the final datasets used for the reported comparisons.

Task	Operator / solver	Condition and target	Normalization	Train/eval.
Radiosity	Fredholm integral equation; Monte-Carlo progressive radiosity	$\hat{\boldsymbol{\ell}}, \rho_d \mapsto B$	Per-scene z-score	480/120
Acoustics	Exterior Helmholtz; Burton–Miller BEM	$\hat{\boldsymbol{k}}_{\mathrm{inc}}, ka \mapsto \lvert p_{\mathrm{tot}} \rvert$	Global training-set z-score	400/100
Electrostatics	Exterior Laplace; single-layer BEM	$\hat{\boldsymbol{E}}_{\infty} \mapsto \sigma$	Signed log and per-scene z-score	160/40

D.1 RADIOSITY

Problem and solver. We model diffuse global illumination with the radiosity equation (Goral et al., 1984; Cohen et al., 1988). For surface patch i,

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j, \tag{15}$$

where B_i is outgoing radiosity, E_i is emission, ρ_i is diffuse reflectance, and F_{ij} is the geometric form factor. We estimate the gather term without assembling the form-factor matrix. For every non-emissive patch, 1,024 cosine-weighted hemisphere rays are traced to their first intersections; the hit indices are reused across progressive bounce iterations. Rays that leave the scene contribute zero. The iteration count is chosen per scene as

$$K = \left\lceil \frac{\log(10^{-3})}{\log \rho_d} \right\rceil, \tag{16}$$

which bounds the remaining geometric-series contribution for the sampled reflectance. In the generated corpus, K ranges from 12 to 35.

Corpus. Each scene contains a closed Cornell enclosure, a square ceiling emitter, and one ShapeNet object. The emitter is a boundary condition and is excluded from the prediction nodes. Light direction and diffuse reflectance form the model condition, $\mathbf{c} = (\hat{\boldsymbol{\ell}}, \rho_d)$. The object pose, emitter position and size, and light intensity are randomized during generation. Because emitter intensity and size are not supplied to the model, the physical radiosity is standardized independently within each scene; the inverse-transform statistics are retained for visualization.

The main corpus contains 600 distinct objects and one scene per object, split by geometry into 480 training and 120 evaluation scenes. Native scenes contain approximately 52,000–304,000 non-emissive panels; training samples 50,000 panels per scene. The *Radiosity control* in Table 3 instead contains 200 scenes of one Stanford bunny, split 160/40. It uses the same solver and scene randomization but tests condition and pose interpolation rather than transfer to unseen geometry.

Emitter information and model inputs. The emitter position follows the supplied light direction rather than an independently sampled hidden position. For enclosure side length L and square-emitter side length a, its ceiling centre is constructed as

$$\mathbf{r}_{\mathrm{light}} = \mathrm{clip}\Big((L/2, L/2) + 0.22L(\hat{\ell}_x, \hat{\ell}_z),\ a/2 + 0.02L,\ L - a/2 - 0.02L \Big), \tag{17}$$

with componentwise clipping to keep the patch inside the ceiling. Emitter faces are excluded from both the exported node geometry and prediction targets. Their spatial footprint therefore appears as the omitted ceiling region in the non-emissive scene point set; a is not a separate condition channel. The model receives this scene geometry together with light direction and reflectance.

Validation. The Monte-Carlo form factors were checked against four closed-form problems. Coaxial parallel squares and perpendicular rectangles match analytic form factors to 0.041% and 0.040%, respectively. A closed cube with uniform emission and reflectance matches its truncated geometric-series solution to machine precision, while a non-uniform concentric-sphere problem matches the corresponding two-surface solution to 0.011%. For the generated corpus, the measured residual from adding 25 further bounce iterations is at most 0.140%, and no generated closed scene has a nonzero ray-escape fraction.

D.2 ACOUSTIC SCATTERING

Problem and solver. A rigid body is illuminated by a unit-amplitude incident plane wave,

$$p_{\mathrm{inc}}(\mathbf{x}) = \exp\Big(ik\hat{\boldsymbol{k}}_{\mathrm{inc}}^{\mathsf{T}}(\mathbf{x}-\mathbf{x}_0)\Big). \tag{18}$$

The total pressure $p_{\mathrm{tot}} = p_{\mathrm{inc}} + p_{\mathrm{sc}}$ satisfies the exterior Helmholtz equation, the sound-hard boundary condition, and the Sommerfeld radiation condition,

$$(\nabla^2 + k^2)p_{\mathrm{tot}} = 0, \qquad \partial_n p_{\mathrm{tot}} = 0 \quad \text{on } \Gamma. \tag{19}$$

The regression target is the surface total-pressure magnitude $|p_{\mathrm{tot}}|$. We use a constant-panel, centroid-collocation Burton–Miller boundary-element formulation with coupling $\beta = +i/k$ (Burton & Miller, 1971). This combined formulation avoids the fictitious resonances of the exterior Neumann problem. Near interactions use distance-adaptive float64 quadrature, while far interactions and the dense solve use complex64 computation. Raw multi-component meshes are converted to their watertight union boundary and remeshed to approximately 20,000 panels.

Corpus. The final corpus contains 500 accepted, distinct geometries with one acoustic condition per geometry and a geometry-disjoint 400/100 split. The nondimensional frequencies use $ka \in \{1, 2, 4, 8\}$, with incident directions and frequency bins balanced within each split. The condition vector contains $\hat{\boldsymbol{k}}_{\mathrm{inc}}$ and a monotone encoding of ka. Unlike radiosity and electrostatics, the stored target is raw $|p_{\mathrm{tot}}|$; a global z-score computed from the training split is applied by the learning pipeline and inverted before evaluation.

Each geometry is accepted only after mesh and solve checks, including outward orientation, static operator identities, a relative linear-system residual below 3×10^{-4}, a condition estimate below 10^5, and rejection of strongly concentrated resonance-like fields. These gates determine the final 500 cases; rejected candidates are not replaced across the train/evaluation boundary.

Validation. On a rigid sphere with approximately 12,000 panels, the predicted surface $|p_{\mathrm{tot}}|$ differs from the Mie-series solution by less than 0.8% at ten tested frequencies, including interior-resonance locations. At a resolved resonance, the uncombined boundary integral equation incurs 24.3% error, whereas the Burton–Miller solve remains at 0.36%. Mesh-refinement tests at $ka = 4$ and $ka = 8$ give empirical convergence orders of 1.12 and 1.28. On the thinnest accepted geometry, complex64 and complex128 solves differ by 3.9×10^{-4} in relative L^2 error of the target magnitude.

D.3 ELECTROSTATICS

Problem and solver. Each scene contains a grounded ShapeNet conductor in a uniform applied field $\boldsymbol{E}_\infty$. Eighty percent of scenes additionally contain an oblate floating conductor with unknown constant potential V_s and zero net charge. With gauge origin $\mathbf{x}_0$, the potential is represented as

$$\phi(\mathbf{x}) = -\boldsymbol{E}_\infty^{\mathsf{T}}(\mathbf{x}-\mathbf{x}_0) + \int_\Gamma \frac{\sigma(\mathbf{y})}{4\pi\|\mathbf{x}-\mathbf{y}\|}\,\mathrm{d}A_{\mathbf{y}}, \tag{20}$$

with $\phi = 0$ on the grounded body, $\phi = V_s$ on the shield, and $\int_{\Gamma_s} \sigma\,\mathrm{d}A = 0$ on the floating component. A constant-panel single-layer BEM collocated at triangle centroids solves the resulting augmented linear system for surface charge and V_s. Typical scenes contain approximately 8,000 body panels and, when present, 2,000 shield panels. Surface charge is predicted on both conductors.

Corpus. The final dataset contains 200 distinct geometries with one applied-field direction per geometry and a geometry-disjoint 160/40 split. Failed candidate scenes are replaced within their assigned split if the condition estimate exceeds 10^8, the potential reconstruction error exceeds 10^{-2}, the conductors intersect, or the transformed label exhibits an extreme tail. The model condition is the unit applied-field direction $\hat{\boldsymbol{E}}_\infty$.

Surface charge is compressed and normalized per scene. Defining

$$s = \max(\mathrm{median}\,|\sigma|, 0.1\,\mathrm{quantile}_{0.95}\,|\sigma|), \qquad t = \mathrm{sign}(\sigma)\log(1 + |\sigma|/s), \tag{21}$$

the stored target is $y = (t - \mu_t)/\sigma_t$. The three inverse-transform constants are retained for physical-space visualization. Reported relative L^2 errors are computed in this normalized target space.

Condition encoding and existing control. The physical condition is the unit field direction. The fourth condition slot has no additional electrostatic variable because the field magnitude is fixed. We set this auxiliary slot to 0.3 for the CFD-pretrained model to keep it within the range seen by the source speed channel. Scratch and GeoPT use 0.0, retaining their baseline encoding. GeoPT's synthetic-dynamics pretraining does not share the CFD speed-channel interpretation. All other inputs, targets, sampling, model architecture, and optimization settings are shared.

Validation. The solver was checked against eight analytic or high-accuracy references. Representative results are summarized in Table 9. The tests cover smooth closed conductors, sharp edges, multiple conductors, the production shield-meshing path, convergence under refinement, charge neutrality, and gauge invariance.

Table 9: **Representative solver-validation results.** These checks are performed before generation of the final cross-operator corpora.

Validation check	Outcome
Radiosity	
Coaxial and perpendicular analytic form factors	0.040–0.041% relative error
Closed cube and concentric spheres	machine precision / 0.011%
Acoustics	
Rigid-sphere Mie series at ten frequencies	$< 0.8\%$ relative L^2 error
Mesh refinement at $ka = 4$ and $ka = 8$	order 1.12 / 1.28
Electrostatics	
Sphere capacitance / uniform-field surface charge	0.076% / 0.079%; order 1.35
Cube / oblate-spheroid capacitance	0.033% / 0.106–0.115%
Floating-neutral charge / gauge shift	0.196%; drift 1.7×10^{-13}

E Additional scaling and data-efficiency results

This section reports the task-level results behind the aggregate trends in Section 6.

E.1 CFD source scale and model capacity

Geometric coverage. Figure 5 expands the aggregate geometry trend in Figure 4(a) into all seven downstream tasks and all six (L, K) sweeps.

Model capacity. The common-corpus comparison in Figure 6 shows that additional depth is most useful when the pretrained representation continues to exploit it. Here Car-Crash, electrostatics, and radiosity improves monotonically from L4 to L32 Other tasks exhibit limited or negative returns. These results show that larger pretrained models help strongly on some downstream problems but do not define a universally improving capacity axis.

E.2 High-fidelity data efficiency

Engineering field benchmarks. Table 10 gives the selected-depth curves plotted in the main figure. The selected configurations are Car-Crash with L32, Aircraft with L32 , NASA-CRM with L16, and LR-matched DrivAerML with L8 . Each curve is matched to full-data Scratch at the same depth. Table 11 reports the targets, interpolation intervals, required fractions, and savings.

Deformable-body tasks. Table 12 extends the same comparison to solid-mechanics pretraining. SphereSimple and SimJEB improve at every measured fraction. DeformingPlate displacement and stress are worse at the 10% point but improve at every measured fraction from 20% onward. At full data, pretraining reduces displacement error by 68.2% on DeformingPlate and 7.8% on SphereSimple, reduces DeformingPlate stress error by 73.5%, and reduces SimJEB displacement error by 37.2%. Using full-data Scratch error as the target, interpolation estimates that DeformingPlate displacement requires 62% of the original trajectories and SimJEB displacement requires 23% of the

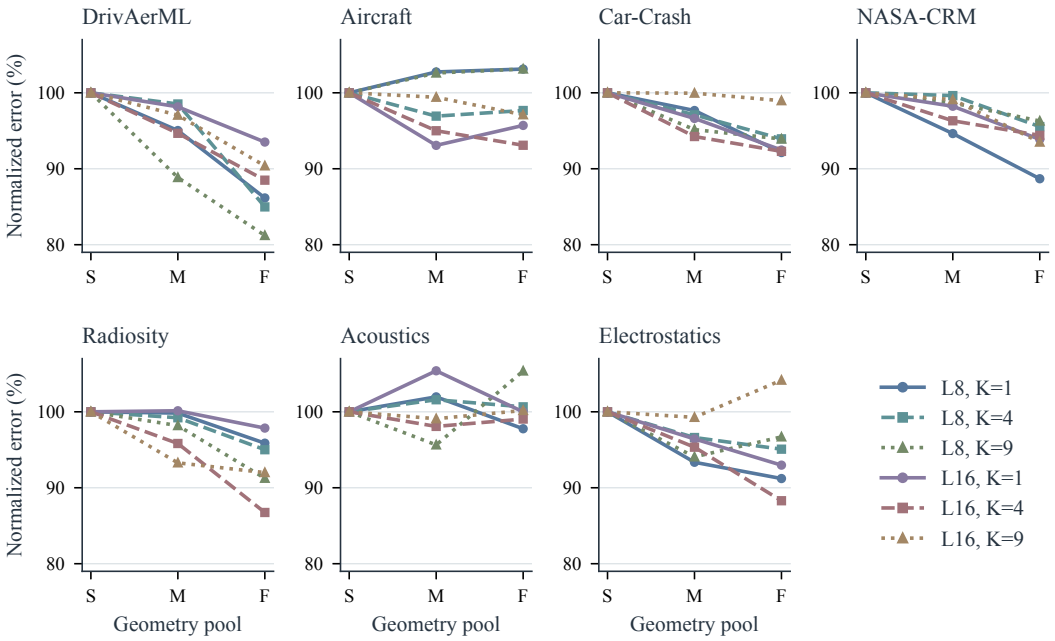


Figure 5: **All six geometry sweeps on each downstream task.** Errors are normalized to each sweep's small-pool error. S/M/F denote the small, medium, and full geometry pools.

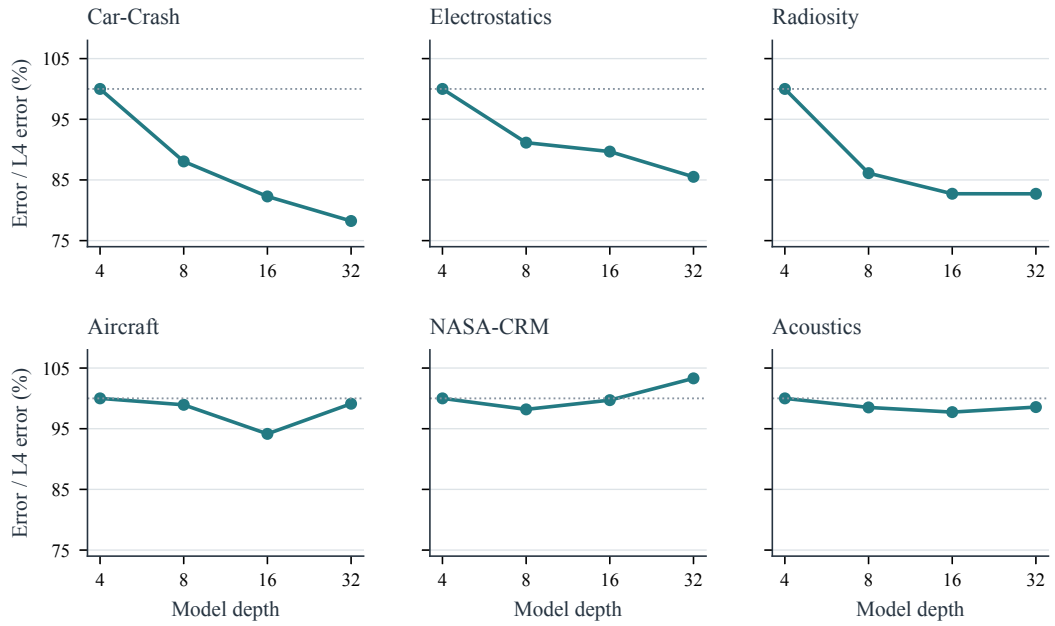


Figure 6: **Capacity trends with a common pretraining corpus.** Errors are normalized to the L4 reference.

original geometry–load pairs, corresponding to savings of 36% and 75%. The data-efficiency benefit extends beyond CFD while depending on the target quantity and data regime.

Table 10: **Engineering curves used for headline label savings.** Relative L^2 error (%) at the selected depth and source checkpoint. Values average available runs of each configuration; most reduced-data cells are single runs. DrivAerML reports LR-matched test pressure error; the other tasks report benchmark validation error. Displayed values are rounded; interpolation uses the unrounded archived values.

Task / depth	Initialization	10%	20%	40%	60%	80%	100%
Car-Crash L32	Scratch	43.7133	47.8634	28.7072	24.3103	21.1425	18.9462
	Ours	23.8393	19.6373	16.3237	15.0747	14.5179	13.8776
Aircraft L32	Scratch	20.9998	8.4031	7.7171	5.9236	6.9507	5.3130
	Ours	15.8697	8.6609	5.5008	4.6658	4.1922	4.2059
NASA-CRM L16	Scratch	49.0971	30.8960	19.3178	15.6898	12.3074	9.3612
	Ours	32.1860	22.2490	13.5549	9.7630	8.5824	7.5108
DrivAerML L8	Scratch	11.1173	6.7121	5.3367	4.9203	4.6666	4.3722
	Ours	6.3906	4.9717	4.4328	4.3142	4.1434	3.9324

Table 11: **Reproducible full-data Scratch matching.** Each target is full-data Scratch at the selected depth. Required fractions are estimated by log–log interpolation in the indicated measured interval; saved fraction equals one minus required fraction.

Task	Depth	Target error (%)	Interval (%)	Required (%)	Saved (%)
Car-Crash	L32	18.94622	20–40	22.9	77.1
Aircraft	L32	5.31302	40–60	43.6	56.4
NASA-CRM	L16	9.36124	60–80	65.9	34.1
DrivAerML	L8	4.37216	40–60	49.1	50.9

F ADDITIONAL QUALITATIVE RESULTS

Figures 7–11, Figures 12–15, and Figures 16–18 provide additional qualitative comparisons for the three evaluation groups. The engineering examples first compare spatial error patterns in surface-field prediction across Scratch, GeoPT, and our CFD-pretrained model. The deformable-body examples then show predicted deformation trajectories and static structural responses from Scratch and solid-pretrained models. Finally, the cross-operator examples visualize radiosity, acoustic pressure, and electrostatic charge after supervised finetuning from the CFD-pretrained initialization. Within each example, methods are shown with the same geometry, viewpoint, and color scale. These visualizations complement the aggregate quantitative results.

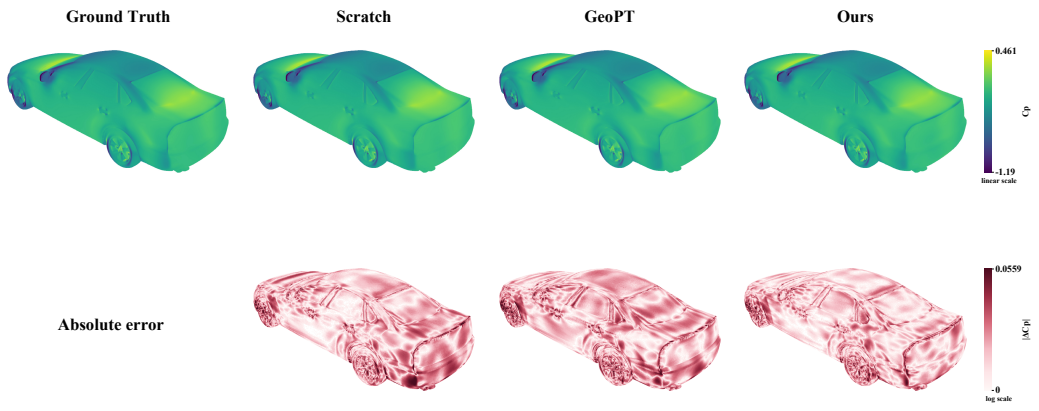


Figure 7: **DrivAerML** C_p Prediction fields and absolute errors.

Table 12: **Complete high-fidelity data-efficiency results on deformable-body tasks.** RMSE for Scratch and solid-pretrained AST models over nested fractions of the downstream training data; lower is better. Displacement entries u are reported as $10^3 \times$ RMSE, while DeformingPlate stress s is reported in its native RMSE scale.

Task / output	Initialization	10%	20%	40%	60%	80%	100%
DeformingPlate u	Scratch	9.442	16.715	12.305	13.000	8.517	6.932
	Ours	13.477	7.394	7.898	7.932	2.576	2.203
DeformingPlate s	Scratch	21669.3	26610.9	25332.0	25245.9	23771.9	25204.4
	Ours	28876.6	17142.0	20749.9	20787.9	7523.7	6668.51
SphereSimple u	Scratch	69.813	80.176	44.787	39.433	33.452	29.741
	Ours	56.724	53.333	38.278	31.718	29.898	27.418
SimJEB u	Scratch	1.534	1.695	1.644	1.454	1.426	0.685
	Ours	1.169	0.752	0.488	0.520	0.454	0.430

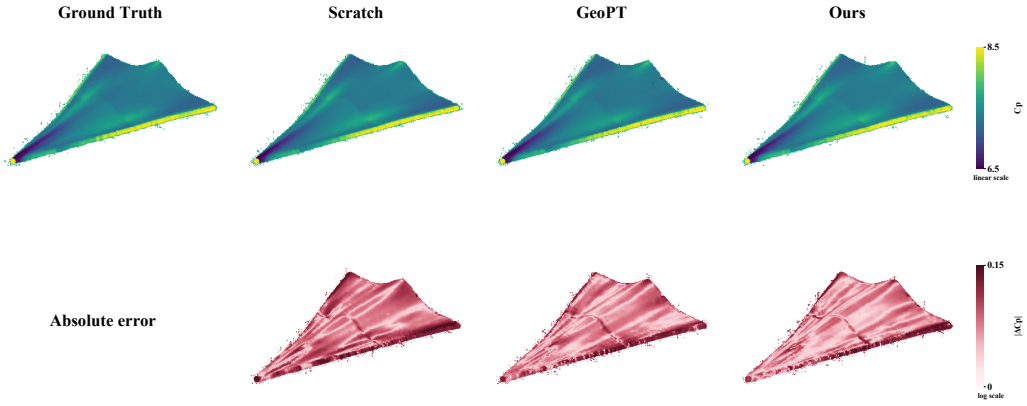


Figure 8: **Aircraft** C_p Prediction fields and absolute errors.

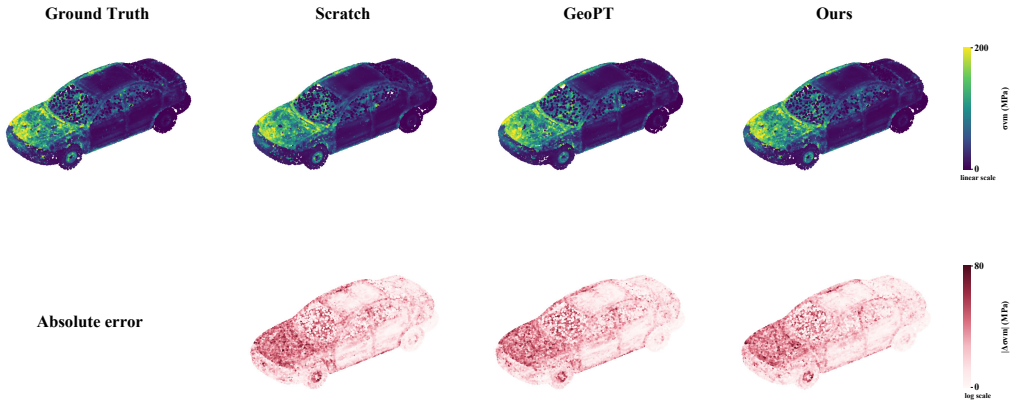


Figure 9: **Car-Crash** σ_{vm} Prediction fields and absolute errors.

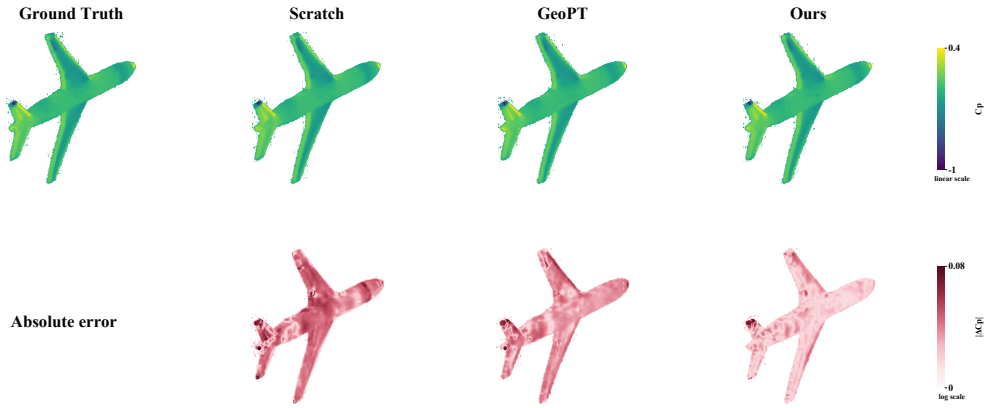


Figure 10: **NASA-CRM** C_p Prediction fields and absolute errors.

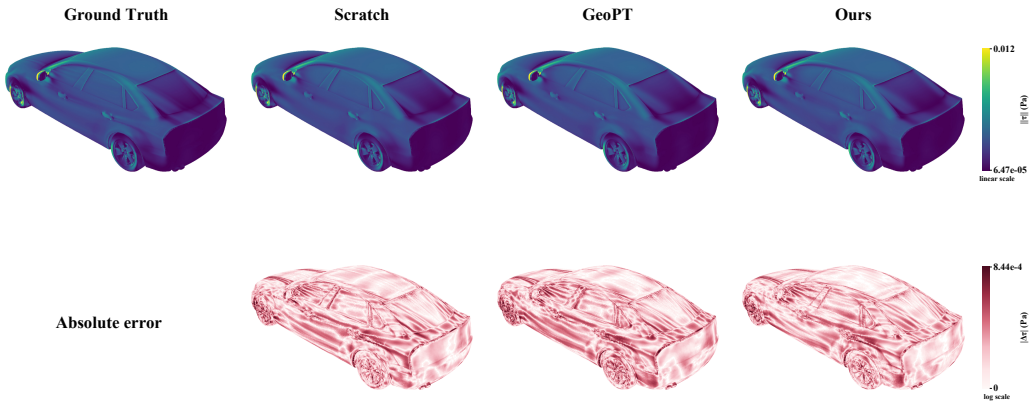


Figure 11: **DrivAerML** $\boldsymbol{\tau}_w$ Prediction fields and absolute errors.

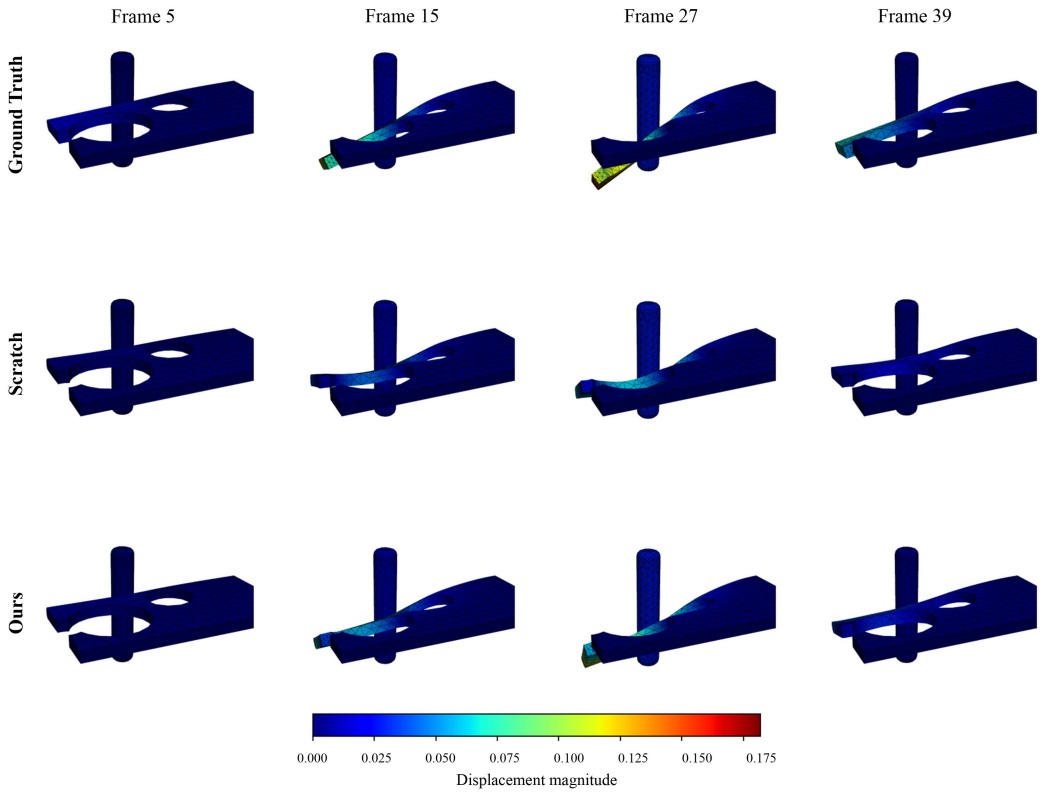


Figure 12: **DeformingPlate displacement rollout.** Columns show four selected frames from the quasi-static deformation sequence; rows compare ground truth, Scratch, and the solid-pretrained model after finetuning. All panels share one displacement-magnitude scale.

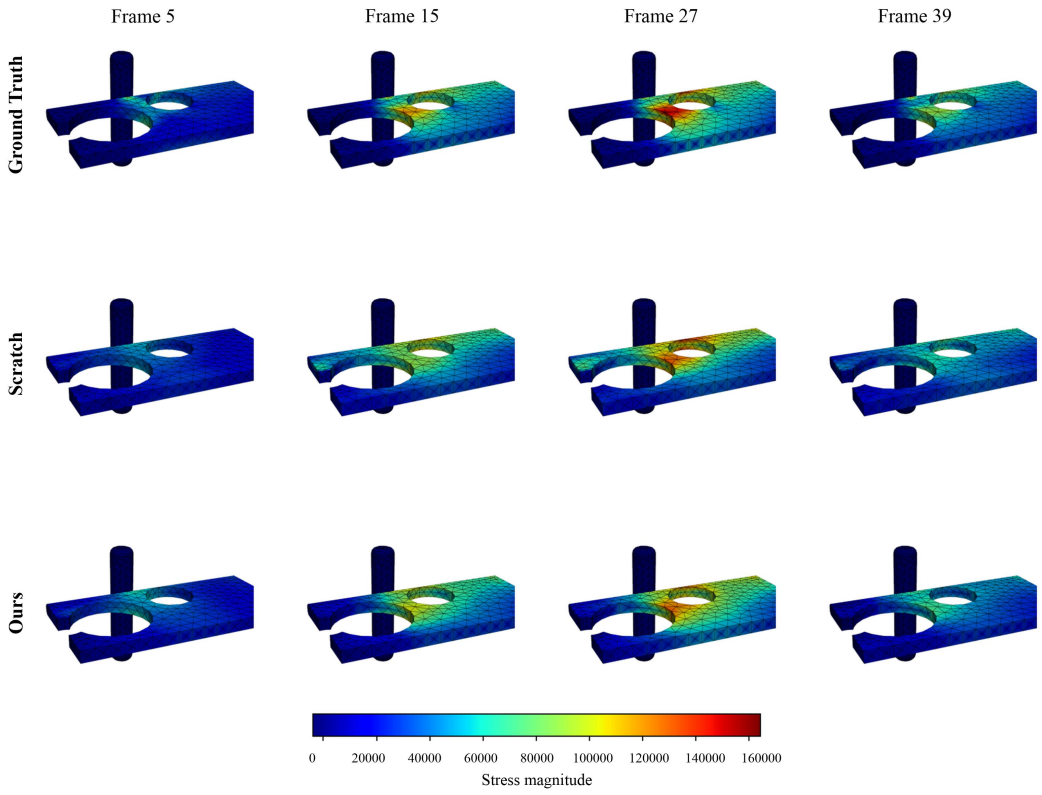


Figure 13: **DeformingPlate stress rollout.** The same trajectory and frames are shown for von Mises stress. Rows compare ground truth, Scratch, and the solid-pretrained model after finetuning under a shared stress scale.

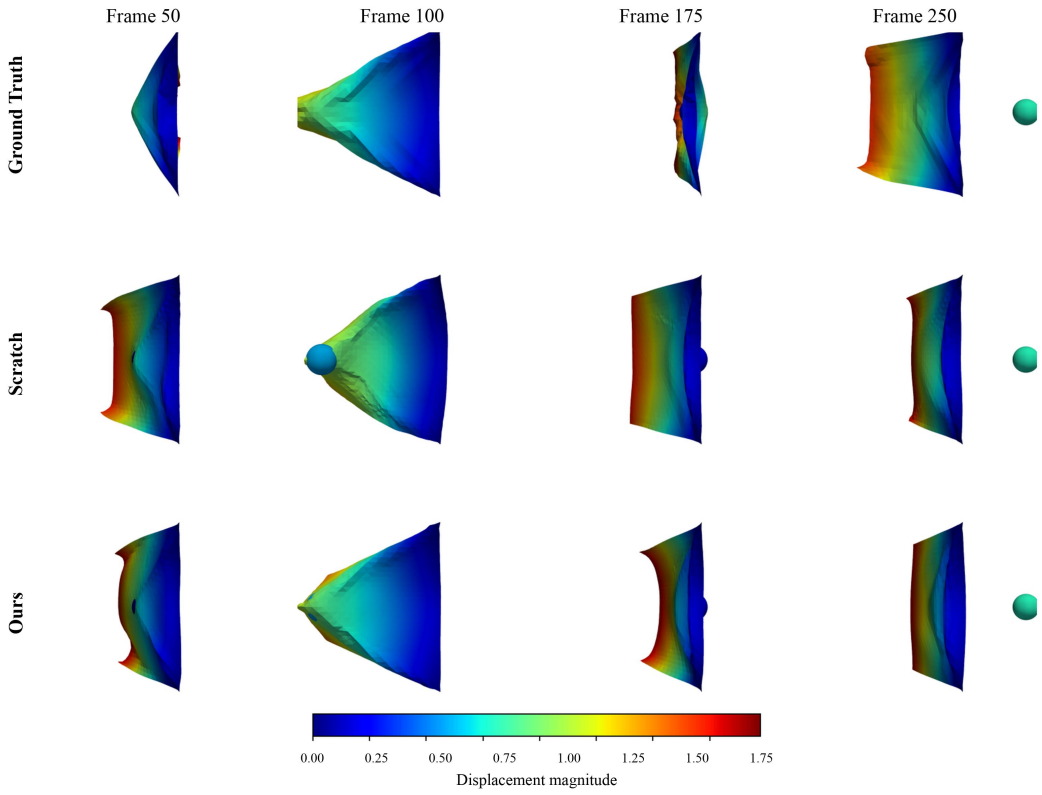


Figure 14: **SphereSimple displacement rollout.** Columns show four stages of the cloth–sphere interaction; rows compare ground truth, Scratch, and the solid-pretrained model after finetuning. Displacement magnitude uses a shared scale across methods and frames.

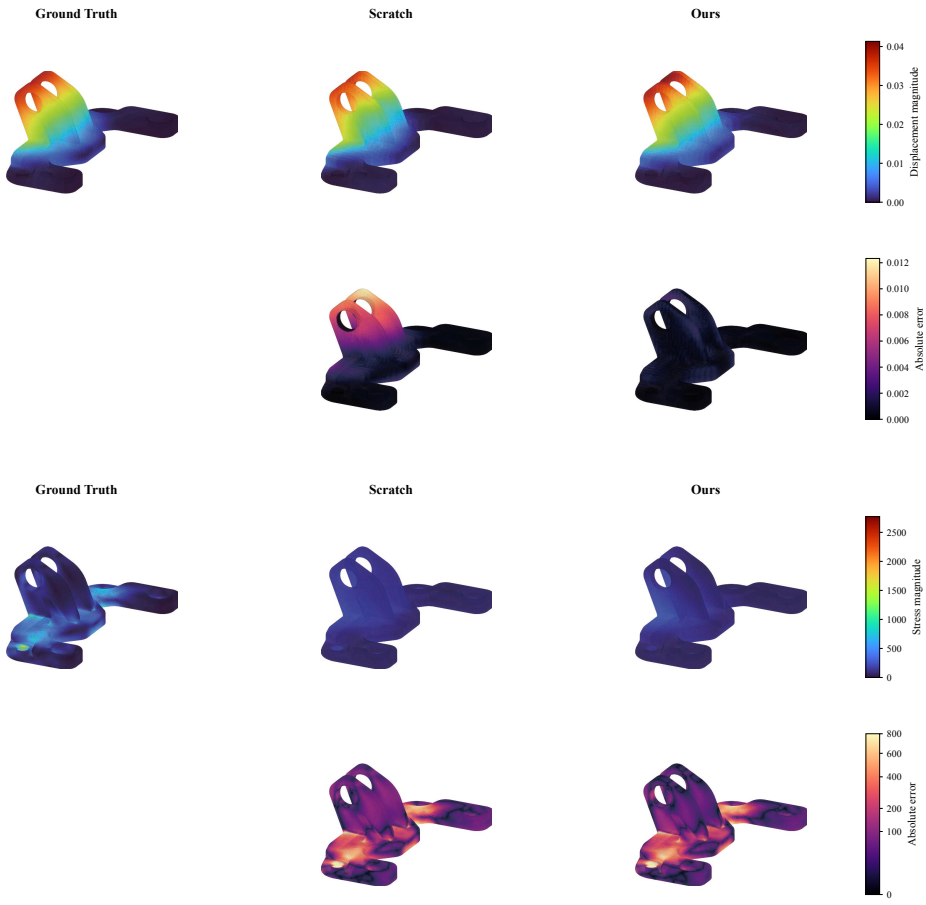


Figure 15: **SimJEB static structural response.** Displacement magnitude (top) and von Mises stress (bottom). Each comparison shows ground truth, Scratch, and the solid-pretrained model after finetuning, together with pointwise absolute-error maps for the two learned models.

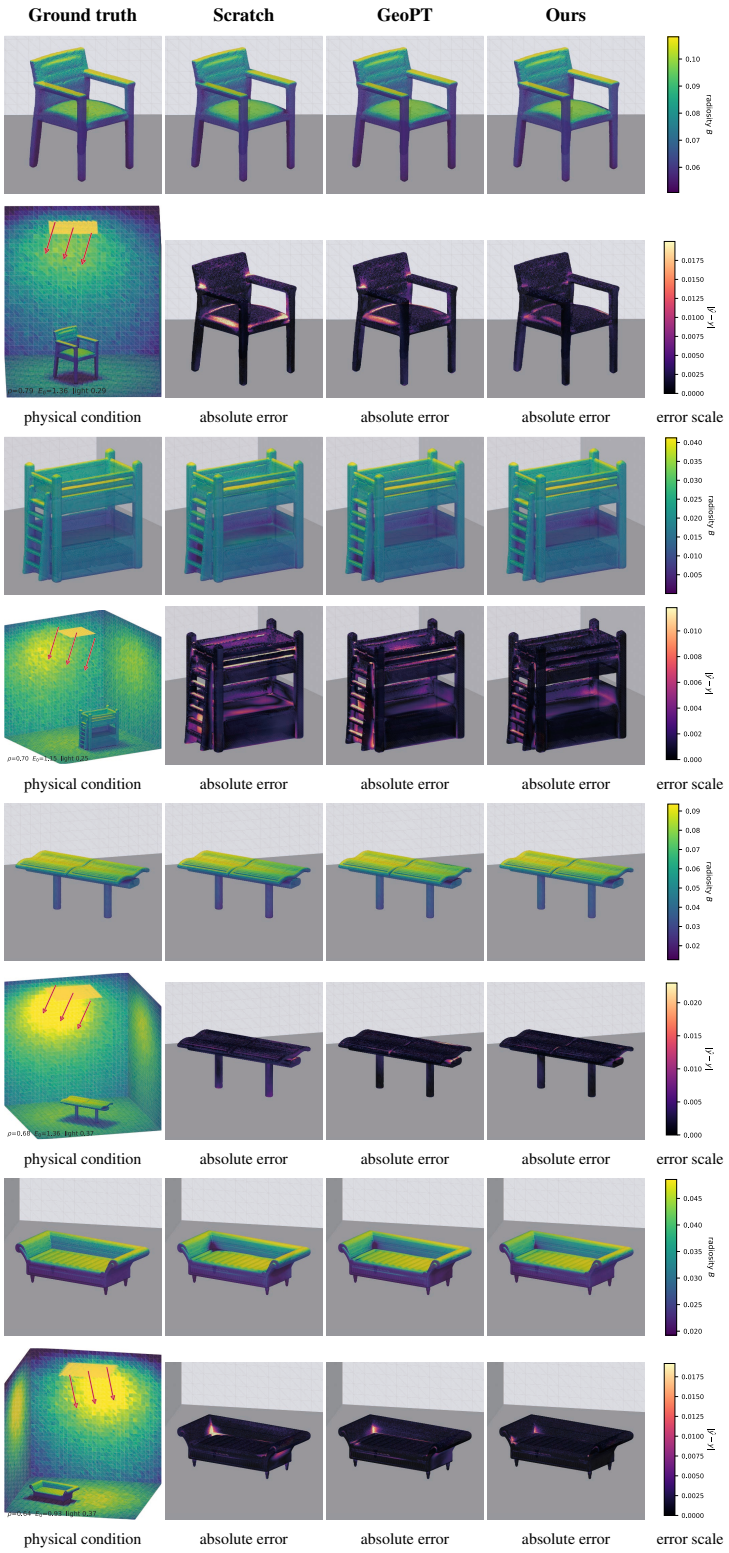


Figure 16: **Radiosity (Cornell-box global illumination), POOL corpus.**

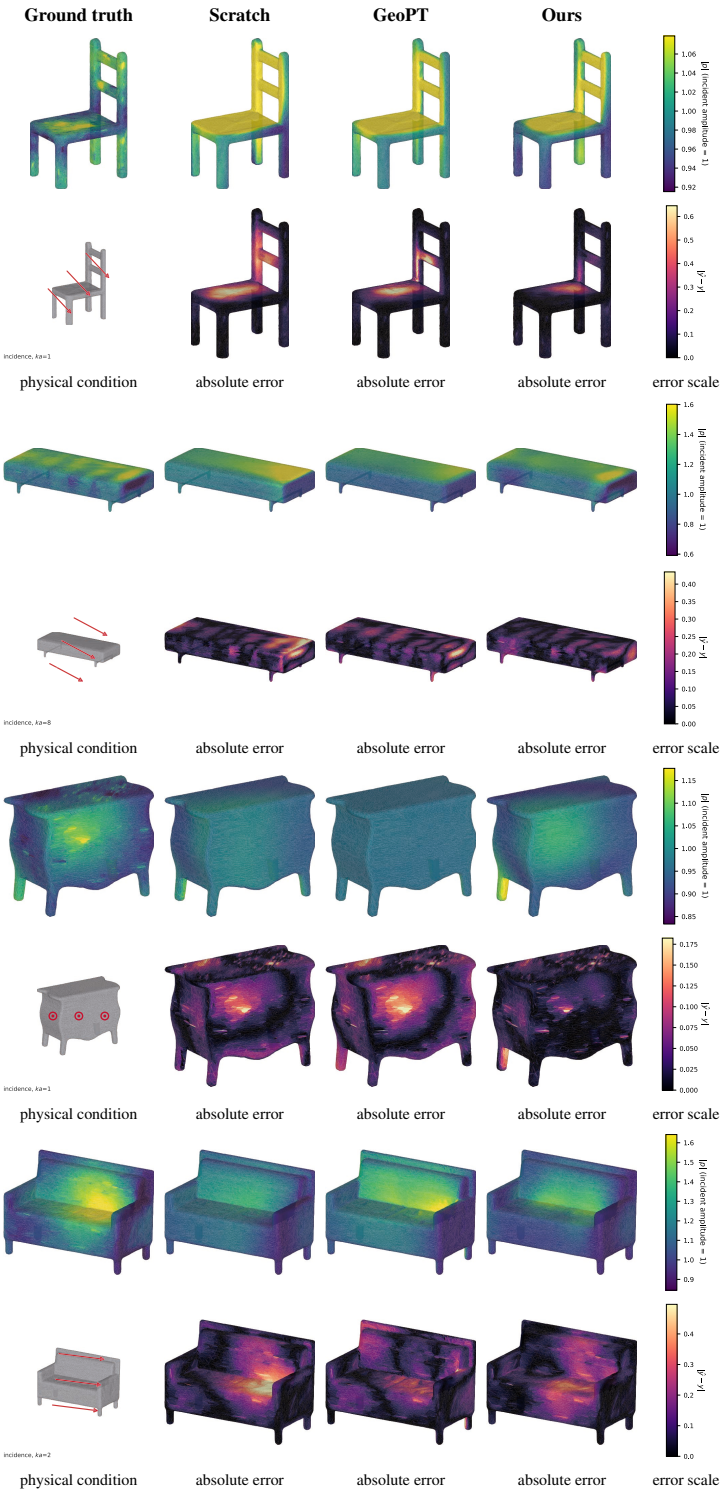


Figure 17: **Acoustics (rigid-body Helmholtz scattering).**

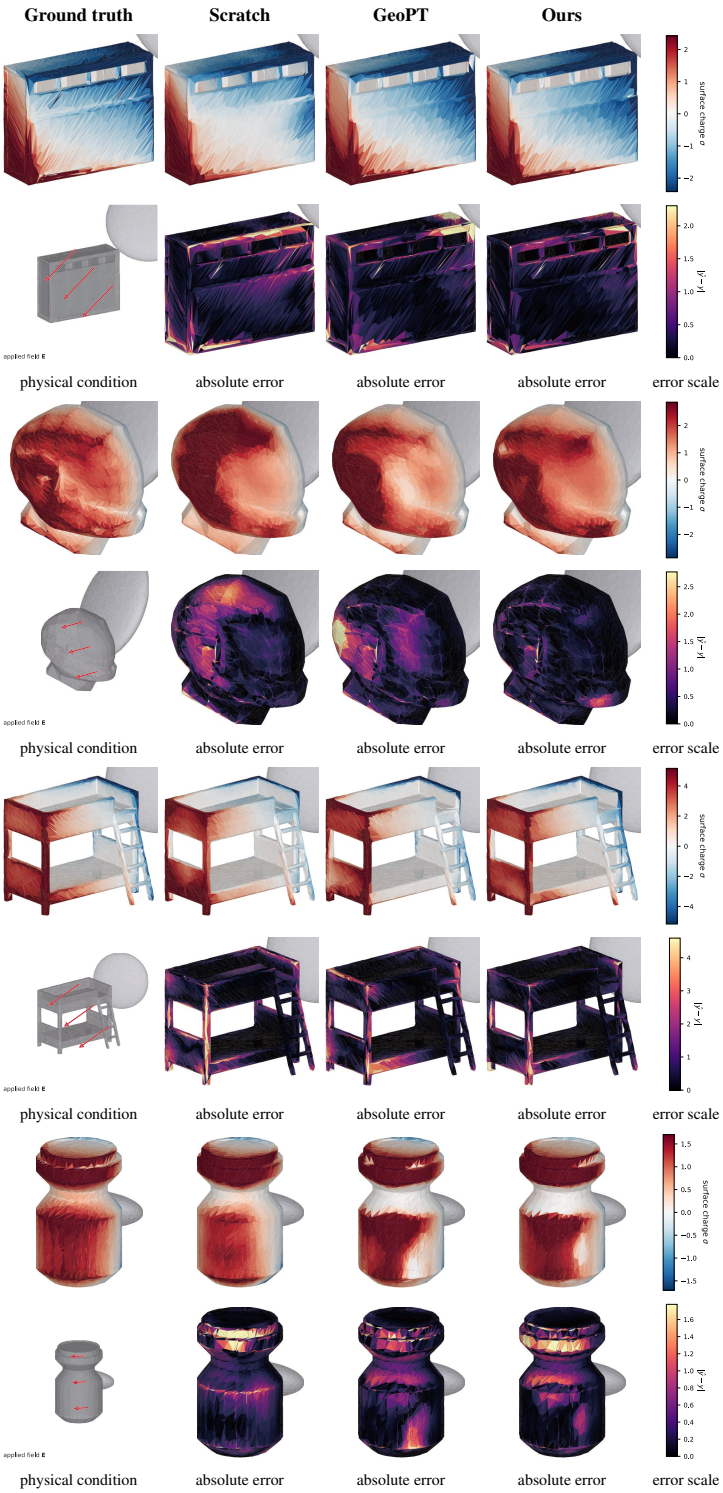


Figure 18: **Electrostatics (induced surface charge).**