BLISS: A DATASET AND BENCHMARK FOR THE BUCKLING OF IMPERFECT SPHERICAL SHELLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine-learning (ML) surrogates are increasingly used to predict how a structure responds to its geometry. However, existing scientific-ML benchmarks rarely use a change in the physical mechanism itself as the source of a distribution shift. Their shifts come from random splits, new initial conditions or held-out ranges of sampled inputs. Here, we introduce **BLISS**, a dataset and benchmark of 8,528 imperfect hemispherical shells simulated with finite elements until collapse, in which an instability creates the shift. A small change in geometry, two defects moving close enough to interact, changes how the shell collapses. Models map the surface with its defects to three outputs of the collapse, the collapse load (knockdown factor), the displacement field and the failure site where collapse starts. Within distribution, as on existing benchmarks, the best models reach a field R^2 of 0.985 and find 98% of clear failure sites. With deeper defects they lose the failure site where defects interact. Trained on shells with far-apart defects and tested on shells with close defects, every model that reads the shell geometry scores below a constant prediction of the average training shell on the field. Moving a single pair of defects from 25° to 10° apart lowers the collapse load by 14% in the solver, while the models predict almost no change. Five dense examples recover most of the field of two models, and the choice between two nearly equal failure sites stays close to chance. Our simulations match eight physical tests within 3.3%. Code, scorer and data-access instructions are available at `https://anonymous.4open.science/w/BLISS-3388/`.

1 INTRODUCTION

Machine-learning surrogates increasingly map the geometry of a part to its physical response, for example the flow around an airfoil or a car (Bonnet et al., 2022; Elrefaie et al., 2024). Many engineering quantities of interest are set by an instability, such as the buckling of a thin wall. Near an instability, a small change in geometry can change the response abruptly, and a surrogate used near an instability must capture that change.

However, existing scientific-ML benchmarks rarely use a change in an instability mechanism itself as the source of a distribution shift. Their test sets differ from training through a random split, new initial conditions or a held-out range of a sampled input, such as the Reynolds number (Takamoto et al., 2022; Ohana et al., 2024; Setinek et al., 2025; Wang et al., 2026). The shift is then set by the sampler (Koh et al., 2021).

Thin shells, such as pressure vessels and launch vehicles, collapse by an instability, and a defect as deep as the wall thickness can halve the pressure a shell carries (Koiter, 1945; Hutchinson, 2016; Lee et al., 2016). The *failure site* is the point where collapse starts. When the defects of a shell are far apart (sparse), each defect acts alone. When two defects come close (dense), they buckle as one region and weaken the shell differently (Derveni et al., 2025a). In learning terms, a small change of the input, two defects moving closer, changes the map from input to output. The generator only sets the defect spacing, and the instability changes the map.

Here, we introduce **BLISS** (Figure 1), a dataset and benchmark that maps the imperfect geometry of hemispherical shells to three outputs of their collapse. These are the *knockdown factor*, the collapse pressure divided by that of the same shell without defects, the displacement field at collapse and the failure site. With BLISS we show that a small change in geometry, two defects moving close enough

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.

to interact, causes an abrupt change in the physics, and that current machine-learning models miss it. Within distribution (B1), like the random splits of existing benchmarks, the best models predict the knockdown factor within 0.6% and find the failure site within 10° on 98% of the shells with one clearly dominant site. With deeper defects (B2), they lose it mainly on dense shells. Trained on sparse shells and tested on dense shells (B3), where defects interact, every model that reads the shape of the shell scores below the *training mean*, a constant prediction of the average training shell, on the field. A controlled edit then isolates the cause. Moving one pair of defects from 25° to 10° apart lowers the knockdown factor by a median of 14% in the solver, while the models predict median changes between −1.1 and +0.3%. The models thus fail twice where defects interact, on the B3 shift and on the controlled edit. Finally, five dense simulations already recover most of the field of FigConvNet and the Transformer, while the choice between two nearly equal candidate sites stays close to chance. Our contributions are the following.

1. **Dataset.** 8,528 nonlinear simulations of shell collapse, with a core of 4,399 shells whose defects are either far apart or close together, 1,870 shells edited one factor at a time for intervention tests and 2,259 shells with one or two defects. The simulations are checked by a mesh study and against eight physical tests.
2. **Protocol.** Three fixed splits (B1, B2, B3) that move from the usual random split to a shift created by the instability, a few-shot variant of B3 and four intervention tests on edited shells. Near-tie scores are compared with how often the solver itself picks the same site (Section 5.4).
3. **Benchmark.** Six geometry models that read the shape of the shell, 20 models that read the list of defects (15 tabular families, three set models and two MLPs), simple learners and references without training, with about 300 trainings of the models that predict the field and 5,194 tabular trials, all scored by one released scorer.

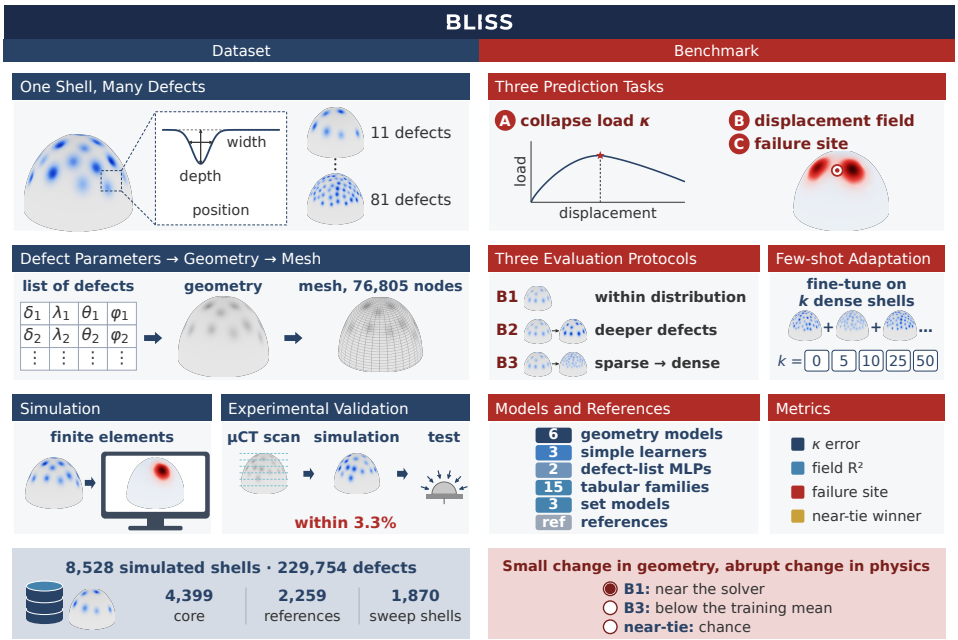


Figure 1: **Overview of BLISS.** Each shell carries many defects, set by their depth, width and position (Section 3). The collapse load κ is the load factor at the first limit point, and the failure site is read from the displacement field at collapse. Simulations of eight scanned and tested specimens match their measured collapse pressures within 3.3%. Sweep shells are the edited shells of Section 4.6. The protocols B1 to B3, the few-shot size k, the models and the metrics are defined in Section 4.

2 Related Work

How existing benchmarks create their shifts. Scientific-ML benchmarks (Table 1) cover problems such as time-dependent PDEs (Takamoto et al., 2022; Ohana et al., 2024; Wang et al., 2026)

and external aerodynamics (Bonnet et al., 2022; Ashton et al., 2024; Elrefaie et al., 2024), and they provide physically consistent fields, often paired with integral quantities such as aerodynamic forces. Most of them build their test set with a random split or by holding out a range of an input chosen by the data sampler, such as the Reynolds number and the angle of attack in AirfRANS or the dominant process parameter in SIMSHIFT. These shifts can be challenging, and AirfRANS reports clearly larger errors under extrapolation, yet the physics stays smooth across them, since its angle of attack was capped at 15° to avoid stall. DrivAerML also holds out cars by drag or by rear flow separation. That split selects test cars by their response over a smoothly varying design space, while in BLISS a small change of the input changes the mechanism itself. RealPDEBench tests models trained on simulations against real measurements (Hu et al., 2026). As Koh et al. (2021) argued, such shifts are often fabricated by samplers instead of arising from the data-generating process, and accuracy in distribution often predicts accuracy under shift (Miller et al., 2021; Baek et al., 2022), although this relation can break or invert (Teney et al., 2023).

Instabilities in existing datasets. Some datasets do contain instabilities or competing outcomes, including the coexisting attractors of a viscoelastic flow in The Well (Ohana et al., 2024), the left or right buckling of the ABC columns (Prachaseree & Lejeune, 2022) and the next broken beam in Voronoi lattices (Karapiperis & Kochmann, 2023). They draw training and test samples from the same generator, however, and release no measure of how close the competing outcomes were. Other structural-mechanics datasets, including shell buckling at scale (Pollet et al., 2026a), and work on multimodal predictors and uncertain labels are discussed in App. K.

What BLISS adds. The generator of BLISS only sets the minimum defect spacing, sparse or dense. Across this split the map from input to output changes, because close defects interact (Section 3), and a controlled edit that moves one pair of defects closer (P3, Section 5.2) isolates that change. BLISS scores the three outputs of this instability (knockdown factor, field and failure site) with a per-shell measure of how close the competing failure sites are and with simulations checked against experiments. To our knowledge, no public dataset combines these. In the benchmarks above the physics is the same on both sides of the split, so their shifts test extrapolation over a smooth map. In BLISS it changes, and Section 5 shows that current models miss the change.

Table 1: Scientific-ML simulation benchmarks and structural-instability datasets compared with BLISS.

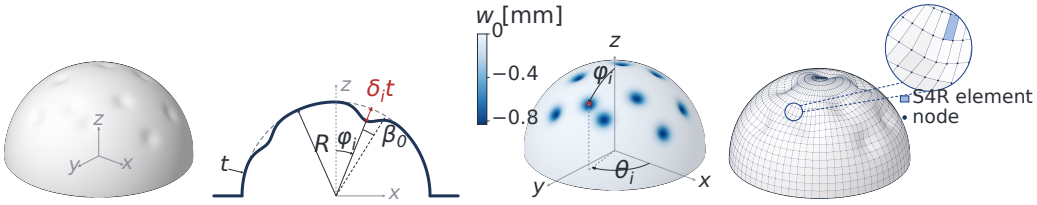
						Data and evaluation axes						
Benchmark	dim.	#sims	#geom	res.	geom.	instab.	sc.+fld	outcome	shift source	exp.	real	margin
PDE and physical-system suites (fluid and PDE solutions)												
PDEBench (2022)	1–3D	0.1–10 k	1	$\leq 128^3$	fixed	✗	✗	✗	random, horizon	✗	✗	✗
The Well (2024)	2–3D	16 sets	1	256^2–64^3	fixed	✗	✗	✗	random	✗	✗	✗
FD-Bench (2026)	2D	1–42 k	1	n/r	fixed	✗	✗	✗	new init. cond.	✗	✗	✗
RealPDEBench (2026)	2–3D	0.4 k	1	n/r	fixed	✗	n/r	✗	sim. to real	✓	✓	✗
Geometry-varying engineering surrogates (fluid and PDE solutions)												
AirfRANS (2022)	2D	1 k	1 k	0.25–0.3 M	mesh+params	✗	✓	✗	input range	✓	✗	✗
DrivAerML (2024)	3D	484	484	160 M	mesh+params	✗	✓	✗	response range	✓	✗	✗
DrivAerNet++ (2024)	3D	8 k	8 k	24 M	mesh+params	✗	✓	✗	random	✓	✗	✗
SIMSHIFT (2025)	2–3D	11.7 k	11.7 k	0.6 k–1.4 M	mesh	✗	✗	✗	input range	✗	✗	✗
Structural mechanics and instability (nonlinear solids, post-buckling, fracture, shell buckling)												
MGN plate (2021)	3D	1.2 k	1.2 k	1.3 k	mesh	✗	✗	✗	random	✗	✗	✗
Mechanical MNIST (2020)	2D	70 k	1	28^2	mesh+params	✗	✓	✗	random, OOD sets	✗	✗	✗
Geo-FNO solids (2023)	2D	2.2 k	2.2 k	1–3 k	mesh	✗	✗	✗	random	✗	✗	✗
ABC columns (2022)	3D	75 k	75 k	n/r	mesh	(✓)	✗	✓	random	✗	✗	✗
Voronoi fracture (2023)	2D	50 k	50 k	162 cells	mesh	(✓)	✗	✓	random	✗	✗	✗
ConcreteShellFEA (2026b)	3D surf.	65 k	65 k	n/r	n/r	(✓)	✓	✗	none	✗	✗	✗
Nonlinear limit-point buckling of imperfect shells (this work)												
BLISS (ours)	**3D surf.**	**8.5 k**	**8.5 k**	**76.8 k**	**mesh+params**	✓	✓	✓	**defect interaction**	✓	(✓)	✓

Columns. The group headings give the type of analysis. In BLISS it is a nonlinear limit point, where the load peaks and then drops. *dim.* is the domain dimension. *#sims* and *#geom* count simulations and geometries. *res.* is the discretisation per sample. *geom.* is a fixed domain, a mesh per sample, or a mesh with its parameters. *instab.* marks a target defined by an instability. *sc.+fld* marks a scalar and a field from one solution. *outcome* marks a scored discrete outcome, such as a failure site. *shift source* says how the test set differs from training. *exp.* marks an experimental anchor. *real* marks released measurements ((✓) used, not released). *margin* marks a released per-sample measure of how close competing outcomes are (the site ratio q of Section 3 in BLISS). ✓ yes, (✓) qualified, ✗ no, n/r not reported by the source. Evidence per cell in Table 32.

3 THE DATASET

To test the claim, the dataset must contain the physical change itself, shells whose defects interact and shells whose defects do not, simulated precisely enough that the change is real.

Each BLISS sample is one imperfect shell (Figure 2), a thin hemisphere with small defects loaded by external pressure. The 4,399 core shells carry every split and task. The release adds 1,870 edited shells, made by rotating, moving or deleting defects of core shells and used for evaluation only, and 2,259 single- and two-defect shells released as tabulated rows. In total, BLISS holds 8,528 simulated shells (Table 9). The release, its access and its datasheet are described in App. B and N.



(a) the shell (b) its parameters (c) its field w (d) its mesh

Figure 2: **Geometry of a BLISS shell.** The input is the surface deviation w at every node of one shared mesh. The targets are the knockdown factor κ and the displacements at collapse. Panel (b) shows the depth $\delta_i t$ and position (θ_i, ϕ_i) of each defect and the half-width β_0 shared by every defect. All shells share the mesh in (d), whose element type is given in App. E.1.

Input. Every shell is the same thin elastic hemisphere, clamped at its rim (App. E.1). It carries 11 to 81 small inward defects of random depth and position, all within 60° of its top, with the defect design of Derveni et al. (2023). The input of a model is the surface deviation w, the distance of each of the 76,805 mesh nodes from a perfect sphere, and all shells share the same nodes. The list of defects, with the depth of each defect in units of the wall thickness t and its position, is also released.

Simulation. A finite-element solver raises the pressure until the shell collapses, which happens at the first peak of the pressure (App. E.1). The three quantities below are read at that peak.

Quantities of interest. Three quantities are predicted for each shell. The *knockdown factor* κ is the collapse pressure divided by the collapse pressure of the same shell without defects. It equals 1 when the defects have no effect and decreases as the shell becomes weaker. The *displacement field at collapse* $\mathbf{U}$ gives the displacement of each node at the peak, three numbers per node. Under pressure the whole shell first shrinks inward almost uniformly. We subtract this *uniform shrink*, the mean inward displacement, and the rest of the field shows a few separate inward deflections. The *failure site* is the node with the largest remaining deflection. The *site ratio* q is the second-largest separate inward deflection divided by the largest one (App. E.1). We call shells with $q > 0.9$ near-ties (**27.5%** of shells) and shells with $q < 0.7$ clear winners. Shells in between enter the field and κ metrics only. Intuitively, a clear winner is a well-separated decision in which one site dominates. A near-tie is analogous to two nearly equal logits, where a small perturbation can flip the winner. The knockdown factor is a single global number, while the field and the failure site describe where and how the shell gives way.

Two defect spacings. The defect spacing carries the physical change. In half of the core shells the defects are far apart (sparse), with any two defect centres at least 25° apart and 11 to 18 defects. In the other half they are close together (dense), at least 10° apart, with 69 to 81 defects. Two defects interact when they lie within about 20° of each other on these shells (Derveni et al., 2023; 2025a), so sparse defects lie mostly outside each other's reach and dense defects mostly within it. The edited shells of Section 4.6 probe the step between the two spacings for one pair of defects, and the spacing at which interaction begins is left open.

Label quality. On a four times finer mesh, κ changes by 0.4% or less on 5 of 6 study shells and by 0.93% on the sixth (App. E.6.1), against model errors of 4.7 to 9.4% on B3 (Section 5.1). On eight real shells that Derveni et al. (2023) scanned and tested, the simulated collapse pressure matches

the measured one within 3.3%, with a **median difference of 1.9%** and **7 of 8** simulations within the spread of ten repeated tests (App. E.6.3). These specimens differ from the released shells, so they check the simulation chain instead of the dataset population. The displacements are checked by mesh refinement only. Two of the three specimens with one clear site in our simulation collapsed beyond 10° from it (App. E.6.4), so we present the failure site as a simulated label, without a claim of experimental validation.

4 BENCHMARK PROTOCOL

The benchmark scores the three outputs of each shell on three protocols that move from a random split to the shift created by defect interaction, and it reads every score against references without training.

4.1 TASK

One sample is one shell. The input is the surface deviation w on a fixed set of 8,192 of the 76,805 mesh nodes, drawn once at random (seed 0), the same for every model and used for all scoring. A model named without a resolution is its 8,192-node version, and other resolutions are given in parentheses. Task A predicts κ. Task B predicts the displacement field, three numbers per node, as in neural operators and mesh-based models. Task C reads the failure site from the predicted field, with no extra output. The models on the list of defects read instead the depth and position of each defect. This input is privileged, because a scanned real shell comes without such a list.

4.2 EVALUATION PROTOCOLS

The three protocols share this task and differ only in how training and test shells are drawn. Their sizes and their composition by defect spacing and site ratio are in Table 7 (App. E.4).

Within distribution (B1). B1 is a random split (seed 2027), the within-distribution control that asks whether the map from w can be learned at all.

Deeper defects (B2). B2 tests extrapolation to the 1,234 shells whose deepest defect exceeds $2.35\,t$, about the deepest 28% of the core. The κ of a single defect changes little beyond about one wall thickness, so B2 mainly tests the field and the site. Its test set holds more dense shells than its training set (62% against 45%), so B2 is also reported per spacing.

Sparse to dense (B3). B3, the shift created by the instability, trains and validates models on sparse shells and tests them on dense shells, where defects interact. B3 also shifts the number of defects, from 11 to 18 in training to 69 to 81 in test, and the intervention test P3 of Section 5.2 separates spacing from count. In a few-shot variant, a B3 model is fine-tuned on $k = 5$, 10, 25 or 50 dense shells from a pool of 200 test shells, which are removed from scoring, and it is scored on the other 2,000 (App. E.4).

4.3 BASELINES AND LEARNED MODELS

The models span the main surrogate families, so a B3 failure cannot be blamed on one architecture or a weak baseline. *Geometry models* read w. DeepONet (Lu et al., 2021), FigConvNet (Choy et al., 2025), MeshGraphNet (Pfaff et al., 2021), a Transformer (Vaswani et al., 2017) and Transolver (Wu et al., 2024) read it on the nodes, and SFNO (Bonev et al., 2023) reads it on a 180×720 image of the sphere. *Models on the list of defects* test how much of each output the defect parameters alone carry. Fifteen tabular families, including XGBoost, and three set models (flat, DeepSets, pairwise) predict κ, and two MLPs predict the field and κ from the 20 or 80 deepest defects (App. G.1). *Simple learners* (a nearest neighbour, PCA with ridge regression and gradient boosting on w) get the same search budget. *References* need no training. The training mean predicts the average training shell, the weakest-link rule predicts κ from the deepest defect alone (App. F), and the depth rules place the failure site at the deepest point of w or pick the deeper of two candidate sites.

Scale. The simulations took about 21,600 core-hours. The models that predict the field took 207 hyperparameter trials, 96 final trainings and 108 few-shot fine-tunings, for about 2,130 GPU-hours including the three controls below, and the tabular models 5,194 search trials and 150 refits in 2,900 CPU core-hours. Three controls retrain models on the full mesh of 76,805 nodes, on 2,048 nodes and, for SFNO, on a 64×128 grid (App. H.7). Verification and validation took 2,400 more core-hours (App. A).

4.4 Training, selection and seeds

So that B2 and B3 test transfer, every choice is made on B1 validation data. All models that predict the field train jointly on field and κ, and each family gets 25 to 30 Optuna hyperparameter trials of 60 epochs on B1. The best configuration is retrained with three seeds and kept fixed for B2 and B3, which choose only their number of epochs. The loss, the search and the selection metric are in App. G.2 to G.4, which also lists the test shells scored by an earlier version.

4.5 Metrics and statistical significance

We report one headline metric per physical output, with a separate winner metric for near-ties. The knockdown factor is scored by its median relative error. The field is scored by the *buckling* R^2, the R^2 of the displacement field after removing the uniform shrink, pooled over nodes and shells, so a fixed mean field can score above zero. We remove the shrink because it follows almost entirely from κ, which is scored separately. The failure site is scored by the *clear-site accuracy*, the share of clear winners whose predicted site lies within 10° of the true one. On near-ties the model must choose between the two candidate sites, the *near-tie decision*. It is scored by the *near-tie winner rate*, the share of near-ties on which the predicted inward deflection is larger at the true failure site than at the second candidate site, with chance at 50%. Confidence intervals and paired tests are in App. G.6.

4.6 Diagnostic and robustness analyses

The protocols measure how far accuracy falls, and the intervention tests show why. The solver also records each shell one load step before and one after the peak, and the change of the three outputs between these states and the peak measures the precision of the labels (App. H.11).

Intervention tests. As in behavioural testing (Ribeiro et al., 2020), we edit test shells one factor at a time, simulate the new answer and apply the trained models unchanged. P1 rotates the defect pattern about the vertical axis, and P2 moves the deepest defect along a meridian. P3 brings one pair of defects from 25° to 10° apart, and P4 deletes defects far from the two candidate sites. P3 isolates the physical change of the claim. A model detects an edit when its predicted change of κ exceeds its own noise under rotation and has the same sign in all three seeds, a rule fixed before the predictions were opened (App. H.10).

5 Benchmark results

Figure 3 shows that accuracy falls from B1 to B3, Figure 4 that the fall follows the physical change, and Section 5.3 what dense data repairs. The complete tables are in App. H.

5.1 Results across the three protocols

Within distribution (B1). On the split that plays the role of existing benchmarks, the geometry models do very well. FigConvNet, MeshGraphNet and the Transformer reach a buckling R^2 between 0.968 and 0.985, while the solver states one step before and after the peak reach 0.993 and 0.966 (Figure 3b). Every geometry model except DeepONet predicts κ with a median error below 1.1%, against 5.13% for the training mean and 2.49% for the weakest-link rule. SFNO and FigConvNet place the failure site within 10° on 98% and 96% of clear winners, against 64% for the deepest point of w. On near-ties the best models reach 65%, above chance and below the 72% at which the solver agrees with itself. XGBoost on the list of defects reaches 0.72% on κ, while the two MLPs on the same list reach a buckling R^2 of only 0.82 and a clear-site accuracy of 71 to 72%. So the global

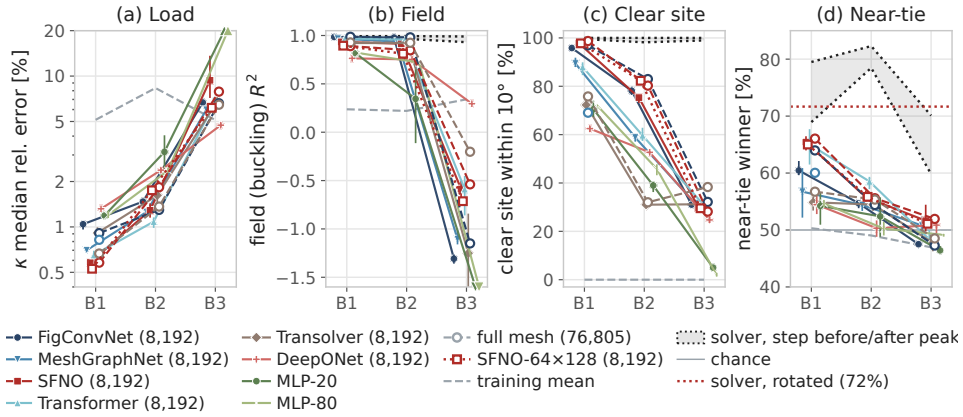


Figure 3: **Results across the three protocols.** Each line is one model, the mean over three seeds, and the vertical bars give the seed range. (a) Median relative error of κ (log scale). (b) Buckling R^2. (c) Clear-site accuracy. (d) Near-tie winner rate, with chance at 50% and the solver's agreement with itself under rotation at 72%. The grey band spans the solver states one step before and after the peak. Triangles mark diverged MLP runs on B3. Dashed lines with open circles are the same models trained on the full mesh (76,805 nodes), and the dotted line is SFNO on a 64×128 grid.

Table 2: **Benchmark results.** B1 plays the role of existing benchmarks, and on B3 every geometry model falls below the training mean. Learned models give the mean over three seeds, and field metrics are taken after removing the uniform shrink. Bold marks the best learned model on 8,192 nodes. Near-tie winner rates are in Figure 3d and App. H. †Reads the list of defects. *SFNO on a 64×128 grid. ‡Deepest point of w. div. diverged. Full-mesh B2 and B3 are reported for FigConvNet, SFNO and Transolver. Models differ in input (w or the list of defects) and output (field and κ, or κ only). The table lists every model that predicts the field and the best κ-only model. Gradient boosting on w, the set models and the other tabular families predict κ only and are in App. H.

	B1 within distribution					B2 deeper defects					B3 sparse to dense				
	MRE κ [%]	R^2 U	RMSE U [mm]	rel. L^2 U [%]	site clear [%]	MRE κ [%]	R^2 U	RMSE U [mm]	rel. L^2 U [%]	site clear [%]	MRE κ [%]	R^2 U	RMSE U [mm]	rel. L^2 U [%]	site clear [%]
references															
training mean	5.13	0.237	0.0339	88	0	8.33	0.221	0.0443	85	0	5.11	0.340	0.0285	82	0
weakest-link rule	2.49	–	–	–	–	2.72	–	–	–	–	5.75	–	–	–	–
depth rules‡	–	–	–	–	64	–	–	–	–	24	–	–	–	–	43
nearest neighbour on w	4.79	0.773	0.0185	50	57	3.88	0.806	0.0221	44	59	4.86	−0.270	0.0396	104	21
PCA and ridge on w	4.26	0.787	0.0179	44	82	6.98	0.537	0.0341	67	54	31.83	−11.864	0.1260	279	31
solver, one step before	–	0.993	–	–	100	–	0.994	–	–	98	–	0.990	–	–	99
solver, one step after	–	0.966	–	–	100	–	0.966	–	–	100	–	0.931	–	–	100
8,192 nodes															
FigConvNet	1.04	**0.985**	**0.0048**	**13**	96	1.47	**0.979**	**0.0073**	**15**	78	6.68	−1.308	0.0534	129	31
MeshGraphNet	0.70	0.968	0.0069	20	89	1.25	0.937	0.0126	25	59	5.99	−1.105	0.0510	119	**32**
SFNO	0.58	0.934	0.0100	26	**98**	1.29	0.915	0.0146	28	75	9.37	−0.587	0.0442	120	30
SFNO-64×128*	**0.53**	0.895	0.0126	28	**98**	1.75	0.812	0.0218	38	**82**	6.16	−0.713	0.0460	125	30
Transformer	0.65	0.969	0.0068	17	88	**1.08**	0.957	0.0104	21	60	6.45	−0.589	0.0442	119	**32**
Transolver	0.91	0.922	0.0108	27	72	1.62	0.921	0.0141	30	31	6.80	−1.251	0.0525	138	31
DeepONet	1.32	0.764	0.0189	51	62	2.38	0.752	0.0250	53	53	**4.72**	**0.295**	**0.0295**	**82**	25
full mesh, 76,805 nodes (scored on the 8,192)															
FigConvNet	0.91	0.987	0.0045	12	97	1.30	0.981	0.0069	14	83	6.60	−1.150	0.0515	123	32
MeshGraphNet	0.82	0.923	0.0107	29	69										
SFNO	0.58	0.890	0.0129	30	99	1.83	0.847	0.0196	35	80	7.89	−0.539	0.0434	119	28
Transolver	0.67	0.928	0.0104	25	76	1.36	0.925	0.0137	29	32	6.46	−0.202	0.0383	104	38
list of defects (parametric input)															
MLP-20†	1.19	0.816	0.0166	43	71	3.14	0.345	0.0394	67	39	div.	div.	div.	div.	5
MLP-80†	1.20	0.817	0.0166	43	72	2.36	0.720	0.0265	55	46	div.	div.	div.	div.	2
XGBoost†	0.72	–	–	–	–	1.69	–	–	–	–	6.50	–	–	–	–

knockdown factor is easier to learn than the local field and failure site (App. J). B1 still separates the models, with a buckling R^2 from 0.76 to 0.985.

Deeper defects (B2). This intermediate test is more challenging, and its losses already concentrate on shells whose defects interact. The five best models keep a buckling R^2 of 0.92 to 0.98 and a κ error of 1.1 to 1.6%, and their clear-site accuracy drops to 31 to 78%. Per spacing, they find the site on all 11 sparse clear winners of the test set and on 29 to 77% of the dense ones, against 56 to 97%

on the dense shells of B1 (Table 21). The larger share of dense shells in the B2 test set explains part of the drop, and worse predictions on the dense shells themselves explain the rest.

Sparse to dense (B3). When the shift is created by the instability, the models fail. Every geometry model scores below the training mean on the field, with a buckling R^2 from -1.31 to 0.29 against 0.340, so predicting the average training shell would be more accurate. The failure is widespread rather than driven by a few shells, with 60 to 79% of B3 shells having negative per-shell R^2 for the five best models (App. H.9). On κ, among the geometry models only DeepONet beats the training mean, with 4.7% against 5.11%, and the others reach 6.0 to 9.4%. Clear-site accuracy falls to 25 to 32%, and the near-tie winner rate lies between 48 and 52%, close to chance. The ranking of the models also inverts. On the field, FigConvNet is the best geometry model on B1 (0.985) and the worst on B3 (-1.31), while DeepONet is the weakest on B1 (0.764) and the least wrong on B3 (0.295). On κ, SFNO is among the most accurate on B1 (0.58%) and the least accurate on B3 (9.37%). Accuracy within distribution therefore fails to predict accuracy under this shift, and the link reported by Miller et al. (2021) inverts here, as Teney et al. (2023) showed it can (Section 2). More input resolution leaves this failure in place. On the full mesh, FigConvNet, SFNO and Transolver reach a buckling R^2 of -1.15, -0.54 and -0.20, and SFNO-64×128 with its own search reaches -0.71 (Tables 2 and 28). All stay below the training mean. The failure is one of transfer. On dense B1 test shells, whose training set also holds dense shells, geometry models reach a buckling R^2 of 0.86 to 0.98 and κ errors of 1.5 to 2.3%, while five dense examples recover most of the field accuracy lost on B3 (Section 5.3). Yet XGBoost and the MLPs on the privileged defect list fail too (Table 2), while set models trained only on the one- and two-defect shells reach median κ errors of 21 to 46% on the core (App. J).

Why the predictions degrade. A model trained on sparse shells has only seen defects that do not interact. Collapse starts at the deepest defect on all 94 sparse B1 test shells with a clear winner, so the models can learn the shortcut "collapse starts at the deepest defect". This rule holds on only 38% of the clear winners among dense shells, because neighbouring defects buckle together. The models extrapolate the non-interacting response to shells where it no longer holds, and all three outputs degrade at once (App. H.2). The field error grows as the second site strengthens. For $q < 0.5$, three models keep a positive median per-shell R^2, with medians falling to -1.0 to -1.4 for $q > 0.5$ (App. H.9). The field even falls below the training mean, because a sharp deflection at the wrong defect adds error twice, where it is predicted and where the true one is missing, while the smooth mean misses the true one only once. With the site wrong on about 70% of clear winners, the sharp prediction loses. On κ the models reach 4.7 to 9.4%, around the 5.75% of the weakest-link rule, which reads only the deepest defect, as the shortcut predicts.

5.2 Moving one pair of defects closer

B3 changes the number and the spacing of the defects together, and dense shells collapse more often near the edge of the 60° cap in which defects are placed, a cap chosen so that the clamped rim leaves κ unchanged (Derveni et al., 2023). The intervention test P3 changes the spacing of one pair and keeps the number of defects (Figure 4). For the shell of Figure 4a, κ stays at 0.434 down to 12.5° and drops to 0.373 at 10°. Its failure site jumps by 56° to a point between the two defects (Figure 4b). Over the 100 shells, the median drop of κ at 10° is 14%. The B1-trained models predict median changes between -1.1 and $+0.3$%, and none recovers half of the true drop on more than 3% of the shells (Figure 4c). The drop appears only between the two closest sampled spacings, 12.5° and 10°, while the predictions remain flat, with the exact onset of interaction left open. B3 shows the failure, and P3, run on B1 models, identifies defect interaction as one mechanism behind it. A control with matched defect counts would separate the two completely.

The failure site shows the same effect. At 10° the solver's failure site lies more than 2° from the centre of every defect on 97% of the shells, because the two defects now buckle as one region. The B1-trained models place their site off every defect on 7 to 33% of the shells, except SFNO with 62%. Models trained on sparse shells (B3) give 4 to 34%, except SFNO with 76% (Figure 4d). The models have learned that collapse starts on a defect, which holds only on sparse shells. The B1 models saw dense shells in training and still respond to one close pair as to a sparse shell. Removing far defects points the same way (App. E.5). In sparse shells κ barely moves (median at most $+1.4$%). In dense shells, keeping 25 to 75% of them lowers κ (median -2.2 to -4.8%, down to -19.4%), and in 92

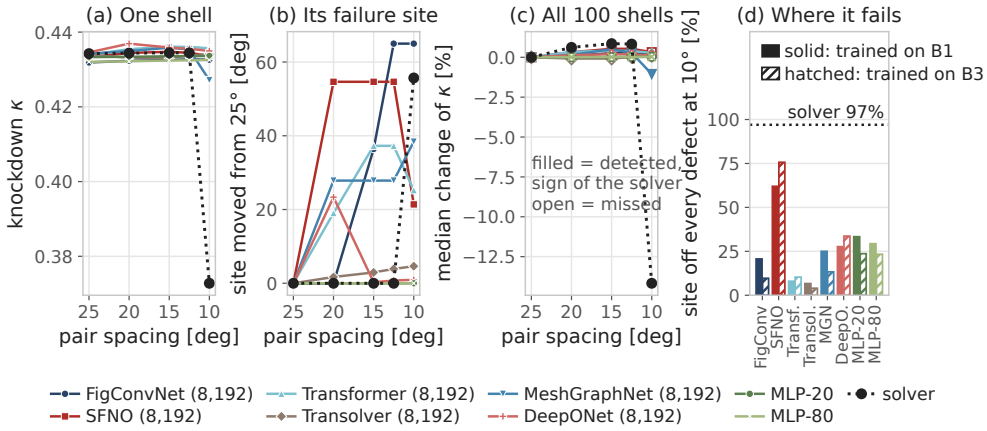


Figure 4: **One pair of defects moved closer (P3).** In 100 sparse B1 test shells, one pair of defects is moved from 25° to 10° apart and every other defect is kept. (a) κ of the shell with the median simulated drop, from the solver and from each B1-trained model (8,192 nodes, mean over three seeds). (b) Distance of its failure site from its site at 25°. (c) Median change of κ over the 100 shells. Filled markers pass the detection rule fixed before the test (App. H.10) with the sign of the solver. SFNO and the Transformer predict a rise at 10° and are drawn open. (d) Share of shells whose failure site at 10° lies more than 2° from the centre of every defect. Solid bars are models trained on B1, hatched bars are models trained on sparse shells (B3), and the dotted line is the solver (97%).

of the 98 edits with a drop above 1% collapse leaves both candidate sites, usually for a far defect that lost its neighbours.

5.3 Few-shot adaptation on dense shells

The B3 models saw no dense shell, and the field adapts quickly once they do. With only five dense shells FigConvNet and the Transformer already reach a buckling R^2 of 0.76 and 0.69 on average, up from -1.31 and -0.59, and 0.89 and 0.82 with fifty. SFNO stays below the training mean, at -0.38. The near-tie winner rate of all three stays between 46 and 52% at every k (Figure 19, Table 24).

5.4 Precision of the solver on near-ties

On a near-tie the two candidate sites deflect almost equally, so its failure site is the least precise label (App. H.11). On B3 near-ties the solver states on either side of the peak pick the peak site on only 60% and 70% of shells (Figure 3d), and rotation reproduces the site of 72% of B1 near-ties, so we read near-tie scores against this reference instead of 100%. The B3 failure rests on the field and on clear winners, where the solver states reach a buckling R^2 of 0.93 to 0.99 and a clear-site accuracy of 99 to 100%.

6 Conclusion

In Bliss, defects moving close enough to interact change the physics abruptly, and current models miss it. They approach the solver within distribution (B1), lose the failure site with deeper defects (B2), fall below the training mean on the field from sparse to dense (B3) and miss the 14% drop of κ in P3, although the B1 models saw dense shells. Five dense examples repair most of the field of two models and leave the near-tie decision close to chance. We believe that a surrogate learns the rule of the regime it sees, and that benchmarks should test, with shifts created by the physics, whether it knows when that rule stops holding.

Scope and future work. By holding the shell family and load case fixed, Bliss is designed to isolate geometry-induced changes in the instability mechanism, so that changes in model performance can be attributed more directly to geometry. Extending this controlled setting to other structures and loads, and learning the near-tie decision, are the next steps (App. L and M).

Ethics statement

BLISS contains no human data. All simulations were performed by the authors. The experimental measurements and micro-CT scans used for validation come from a published study, which we cite, and they are not redistributed. The dataset is licensed under CC BY 4.0 and the code under MIT. We foresee no harmful use. The intended use is research on surrogate models for safer thin-walled structures.

Reproducibility statement

The splits, loader, metric code, baseline implementations and hyperparameter-search results are available from an anonymised code repository, and a sample of the dataset from an anonymised data repository (Appendix B). The full dataset and the prediction file behind every number in this paper are available to reviewers on request and will be public on publication. Section 3 and Appendix N describe the generation pipeline and the conventions needed to regenerate any shell. Appendix G.1 lists every architecture and the selected hyperparameters, Appendix E.6.3 documents the validation, and Appendix E.6.1 the mesh convergence.

References

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

Christina Baek, Yiding Jiang, Aditi Raghunathan, and J. Zico Kolter. Agreement-on-the-line: predicting the performance of neural networks under distribution shift. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1401. arXiv:2206.13089.

Zheren Baizhikova, Uba K. Ubamanyu, Fani Derveni, Roberto Ballarini, Pedro M. Reis, and Jia-Liang Le. A probabilistic buckling model for hemispherical shells with non-interacting localized defects. *Journal of the Mechanics and Physics of Solids*, 208:106468, 2026. doi: 10.1016/j.jmps.2025.106468.

Lucas Beyer, Olivier J. Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with ImageNet?, 2020. arXiv:2006.07159.

Christopher M. Bishop. Mixture density networks. Technical Report NCRG/94/004, Aston University, 1994.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical Fourier neural operators: learning stable dynamics on the sphere. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2806–2823, 2023.

Florent Bonnet, Jocelyn Mazari, Paola Cinnella, and Patrick Gallinari. AirfRANS: high fidelity computational fluid dynamics dataset for approximating Reynolds-averaged Navier–Stokes solutions. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks)*, 2022. doi: 10.52202/068431-1705. arXiv:2212.07564.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

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.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *Algorithmic Learning Theory*, volume 9925 of *Lecture Notes in Computer Science*, pp. 67–82, 2016. doi: 10.1007/978-3-319-46379-7_5.

Dassault Systèmes Simulia Corp. *Abaqus/Standard User's Manual, Version 2024*. Dassault Systèmes Simulia Corp., 2024.

Fani Derveni, William Gueissaz, Dong Yan, and Pedro M. Reis. Probabilistic buckling of imperfect hemispherical shells containing a distribution of defects. *Philosophical Transactions of the Royal Society A*, 381(2244):20220298, 2023. doi: 10.1098/rsta.2022.0298.

Fani Derveni, Arefeh Abbasi, and Pedro M. Reis. Defect–defect interactions in the buckling of imperfect spherical shells. *Journal of Applied Mechanics*, 92(4):041003, 2025a. doi: 10.1115/1.4062774.

Fani Derveni, Florian Choquart, Arefeh Abbasi, Dong Yan, and Pedro M. Reis. The most severe imperfection governs the buckling strength of pressurized multi-defect hemispherical shells. *Mechanics of Materials*, 204:105295, 2025b. doi: 10.1016/j.mechmat.2025.105295.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010.

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 (Datasets and Benchmarks)*, 2024. doi: 10.52202/079017-0016. arXiv:2406.09624.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64 (12):86–92, 2021. doi: 10.1145/3458723.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: a deep neural network with an integrated reject option. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159, 2019.

Fleur Hendriks, Ondřej Rokoš, Martin Doškář, Marc G. D. Geers, and Vlado Menkovski. Equivariant flow matching for symmetry-breaking bifurcation problems. In *NeurIPS Workshop on Machine Learning and the Physical Sciences*, 2025. arXiv:2509.03340.

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 *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. arXiv:2601.01829.

John W. Hutchinson. Imperfection sensitivity of externally pressurized spherical shells. *Journal of Applied Mechanics*, 34(1):49–55, 1967. doi: 10.1115/1.3607667.

John W. Hutchinson. Buckling of spherical shells revisited. *Proceedings of the Royal Society A*, 472 (2195):20160577, 2016. doi: 10.1098/rspa.2016.0577.

Konstantinos Karapiperis and Dennis M. Kochmann. Prediction and control of fracture paths in disordered architected materials using graph neural networks. *Communications Engineering*, 2: 32, 2023. doi: 10.1038/s44172-023-00085-0.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: a benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664, 2021.

Simon A. A. Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R. Ledsam, Klaus H. Maier-Hein, S. M. Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic U-Net for segmentation of ambiguous images. In *Advances in Neural Information Processing Systems*, volume 31, 2018. arXiv:1806.05034.

Warner T. Koiter. *On the stability of elastic equilibrium*. PhD thesis, Delft University of Technology, 1945. Originally *Over de stabiliteit van het elastisch evenwicht*; English translation NASA TT F-10,833 (1967).

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Samuel Lanthaler, Roberto Molinaro, Patrik Hadorn, and Siddhartha Mishra. Nonlinear reconstruction for operator learning of PDEs with discontinuities. In *International Conference on Learning Representations*, 2023. arXiv:2210.01074.

Anna Lee, Francisco López Jiménez, Joel Marthelot, John W. Hutchinson, and Pedro M. Reis. The geometric role of precisely engineered imperfections on the critical buckling load of spherical elastic shells. *Journal of Applied Mechanics*, 83(11):111005, 2016. doi: 10.1115/1.4034431.

Emma Lejeune. Mechanical MNIST: A benchmark dataset for mechanical metamodels. *Extreme Mechanics Letters*, 36:100659, 2020. doi: 10.1016/j.eml.2020.100659.

Zongyi Li, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023. arXiv:2207.05209.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. arXiv:2210.02747.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2946. arXiv:2308.05732.

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. doi: 10.1038/s42256-021-00302-5.

John P. Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7721–7735, 2021.

William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The Well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks)*, 2024. doi: 10.52202/079017-1430. arXiv:2412.00568.

Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, 2018. doi: 10.1016/j.neunet.2018.08.019.

Joshua C. Peterson, Ruairidh M. Battleday, Thomas L. Griffiths, and Olga Russakovsky. Human uncertainty makes classification more robust. In *IEEE/CVF International Conference on Computer Vision*, pp. 9616–9625, 2019. doi: 10.1109/ICCV.2019.00971.

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. arXiv:2010.03409.

Barbara Plank. The "problem" of human label variation: on ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 10671–10682, 2022. doi: 10.18653/v1/2022.emnlp-main.731.

Maxime Pollet, Paul Shepherd, Will Hawkins, and Eduardo Costa. Fast structural analysis of concrete thin-shells using deep learning. *Computers & Structures*, 320:108042, 2026a. doi: 10.1016/j.compstruc.2025.108042.

Maxime Pollet, Paul Shepherd, Will Hawkins, and Eduardo Costa. ConcreteShellFEA: a surrogate modelling dataset for the buckling and stress behaviour of concrete thin-shells. University of Bath Research Data Archive, 2026b.

Peerasait Prachaseree and Emma Lejeune. Learning mechanically driven emergent behavior with message passing neural networks. *Computers & Structures*, 270:106825, 2022. doi: 10.1016/j.compstruc.2022.106825.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4902–4912, 2020. doi: 10.18653/v1/2020.acl-main.442.

Eduard Riks. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures*, 15(7):529–551, 1979. doi: 10.1016/0020-7683(79)90081-7.

Patrick J. Roache. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, Albuquerque, NM, 1998.

Paul Setinek, Gianluca Galletti, Thomas Gross, Dominik Schnürer, Johannes Brandstetter, and Werner Zellinger. SIMSHIFT: a benchmark for adapting neural surrogates to distribution shifts. *arXiv preprint arXiv:2506.12007*, 2025.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 614–622, 2008. doi: 10.1145/1401890.1401965.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks)*, 2022. doi: 10.52202/068431-0117. arXiv:2210.07182.

Damien Teney, Yong Lin, Seong Joon Oh, and Ehsan Abbasnejad. ID and OOD performance are sometimes inversely correlated on real-world datasets. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-3139. arXiv:2209.00613.

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, pp. 5998–6008, 2017.

Haixin Wang, Ruoyan Li, Fred Xu, Fang Sun, Kaiqiao Han, Zijie Huang, Ching Chang, Xiao Luo, Wei Wang, and Yizhou Sun. FD-Bench: a modular and fair benchmark for data-driven fluid simulation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 9894–9905, 2026. doi: 10.1145/3770855.3817497. arXiv:2505.20349.

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

Lingxiao Yuan, Harold S. Park, and Emma Lejeune. Towards out of distribution generalization for problems in mechanics. *Computer Methods in Applied Mechanics and Engineering*, 400:115569, 2022. doi: 10.1016/j.cma.2022.115569.

TABLE OF CONTENTS FOR APPENDICES

A **Reproducibility statement** 15
Regenerating a shell; Compute; Environment (Table 3)

B **URL and links** 16
BLISS dataset; Croissant metadata; Benchmark code

C **Author statement and data licence confirmation** 16
Third-party assets

D **Broader impact** 16
Risks of over-trusting a surrogate; Environmental cost

E **Dataset details** 17
E.1 The shell and the simulation
E.2 The core population (Table 4, Figure 5)
E.3 What a record holds in release 1.0 (Table 5, Table 5)
E.4 Splits (Table 6, Table 7, Table 8, Table 9)
E.5 The single-factor sweeps (Figure 6)
E.6 Verification and validation (Table 10, Figure 7, Table 11, Figure 8, Figure 9, Figure 10, Figure 11)

F **Physics baseline in detail** 27
Step 1, the depth of the worst defect; Step 2, the single-defect curve; Step 3, the prediction; Step 4, does the rule match the data?; Variants (Figure 12)
F.1 The other reference predictors (Table 12)

G **Benchmark details** 29
(Table 13, Table 14)
G.1 Learned models
G.2 Training configuration (Table 15)
G.3 Input configuration
G.4 Model selection on a held-out validation split
G.5 Metric definitions (Table 16, Table 17)
G.6 Why a bootstrap over shells?

H **Full results** 35
H.1 Complete tables per protocol (Table 18, Table 19, Table 20, Table 21)
H.2 Field reconstructions (Figure 13, Figure 14)
H.3 Training-seed spread (Table 22)
H.4 Hyperparameter search (Table 23, Figure 15)
H.5 Training and validation curves (Figure 16, Figure 17, Figure 18)
H.6 Few-shot adaptation on B3 (Figure 19, Table 24)
H.7 Full native mesh and input resolution (Table 25, Table 26, Table 27, Table 28)
H.8 Tabular models (Table 29, Table 30)
H.10 Intervention tests
H.11 Precision of the labels (Table 31)

I **Why does the near-tie decision stay close to chance from sparse to dense?** 51
Near-tie decision within distribution; Near-tie labels on B3; Field and near-tie decision after fine-tuning; Clear winners on B3

J **Does the load need the full geometry?** 52
Knockdown factor from the list of defects; Weakest-link rule on sparse shells; Across the shifts; Composing single- and two-defect shells

K **How does this dataset compare with public benchmarks?** 52
Structural-mechanics datasets; Competing outcomes and label precision in related work (Table 32)

L **Limitations** 55
Failure-site position in dense shells (Figure 24)

M **Open problems** 56
Learning the near-tie decision from few examples; Using the released site ratio; Finding what decides a near-tie; Carrying the thickness

N **Datasheet for Datasets** 56

O **LLM usage statement** 61

A REPRODUCIBILITY STATEMENT

We provide a repository to reproduce every experiment of the paper, with a link to download the dataset (App. B). It holds the split files, the loader, the scorer and the training code of every baseline. During review, the sample of App. B runs the data preparation, the training of every field model, the weakest-link reference and the scorer end to end, and the full dataset, the search journals and every prediction file behind every number in this paper are available to reviewers on request through OpenReview, which keeps both sides anonymous. All of them are public on publication. A new model is scored with one call to the released scorer (App. G.4).

Regenerating a shell. Every released shell is defined by its list of defects, from which w is rebuilt on the shared mesh to 2×10^{-5} relative. A user who solves a shell again must match the rule that picks the first peak of the load, the reference surface of the nodes, the material constants and the solver settings, all listed in App. E.1 and in the datasheet (App. N).

Compute. Table 3 lists the compute of every block. The released simulations took about 21,600 core-hours of finite-element solves. The 4,399 core shells took 12,347 of them, with a median of 2.0 core-hours per solve. Verification and validation took a further 2,400 core-hours. Training and evaluating the benchmark took about 2,130 GPU-hours of training time, about 1,430 on H100 and 635 on A100 cards, the rest on A6000, L40S, A5000/A5500 and A40 cards, and the tabular models took 2,900 CPU core-hours. The allocated GPU time was larger, about 5,540 GPU-hours, of which 3,415 on a national computing centre, 1,071 on a university cluster and about 1,050 on a cloud provider. The cloud share is estimated from activity windows (range 950 to 1,200). Simulations run for exploration and diagnostics that are outside the release are not counted.

Table 3: **Compute.** Simulation rows give the allocated core-hours (elapsed time $\times$ cores) of every scheduler job of each block, with re-runs included, runs that repeat a core shell excluded, and verification and validation listed apart. GPU rows give each run's own training clock (all runs of this version, pruned search trials excluded). For simulations, runs counts the finite-element solves, each done once because the solver is deterministic. The eight specimens were solved at three Young's moduli (1.24, 1.25 and 1.26 MPa), which give the bars of Figure 8. For GPU rows, runs counts the distinct trained runs (tag and seed) behind the hours, and the hours are the sum over the three seeds. Median per solve is the median allocated core-hours of one completed one-shell solve. The sweep cells run several shells per task and have no such median. †Estimate for runs still going.

item	runs	median per solve [core-h]	hours	unit
simulation (Abaqus, 8 cores per solve)				
singles (399)	399	0.87	664	core-h
core (4,399 released)	4,399	2.04	12,347	core-h
doubles (1,860)	1,860	1.41	4,975	core-h
P1 rotation (300)	300	–	613	core-h
P2 meridian (750 released)	850	–	1,222	core-h
P3 pair spacing (500)	500	–	934	core-h
P4 far defects removed (320 released)	400	–	798	core-h
dataset total	–	–	**21,552**	core-h
verification & validation: mesh / flip studies	–	–	1,857	core-h
verification & validation: 8 specimens $\times$ 3 moduli	24	–	12	core-h
verification & validation: near-tie mesh study	80	–	530	core-h
benchmark v2 (GPU)				
B1 finals	39	–	272	GPU-h
B1 finals, second configuration	49	–	237	GPU-h
B2 finals	24	–	146	GPU-h
B3 finals	24	–	101	GPU-h
other	100	–	39	GPU-h
resolution ladder	18	–	118	GPU-h
search	217	–	876	GPU-h
search, second round	145	–	155	GPU-h
few-shot v2 (H100, cells' own timers)	108	–	2.1	GPU-h
SFNO (64×128 grid), pure control (A5500)	3	–	2.2	GPU-h
SFNO-64×128 search and finals (A6000)	28	–	17.6	GPU-h
sweep / rotation inference (H100)†	–	–	0.6	GPU-h
full mesh FigConvNet B2 and B3 (H100)	6	–	160	GPU-h
GPU total	–	–	**2,127**	GPU-h
tabular rail (CPU)				
tabular searches and finals (pre-release)	–	–	2	core-h
tabular searches and finals (v2)	–	–	2,887	core-h

Environment. PyTorch 2.9.1 (CUDA 12.8), PhysicsNeMo 2.2.2 (FigConvNet), torch-harmonics 0.8.0 (SFNO) and Python 3.11. The release carries a `requirements.txt` with these versions.

B URL AND LINKS

URLs that would identify the authors are withheld for anonymous review and will be given in the camera-ready version.

BLISS dataset. During review a sample of the dataset is available from an anonymised repository `https://osf.io/vyr3c/?view_only=4c5fd2f7aae1440e8de3e0bebece4648`. It holds 55 core shells with their fields, 25 records of the intervention test P3, and 25 single- and 25 two-defect shells, in the format of the full release, with the fields on all 76,805 nodes. The full release is available to reviewers on request through OpenReview, which keeps both sides anonymous, and all reported results use it. On publication it will be hosted on the Hugging Face Hub with a DOI, under CC BY 4.0. It holds the core (about 16 GB, the shared mesh, one record per core shell, the metadata and the splits, which is everything the benchmark needs), the 1,870 sweep shells with their fields, and the single- and two-defect shells as tabulated rows.

Project page. A project page with a dataset explorer, the leaderboard and the links to the data and the code is at `https://anonymous.4open.science/w/BLISS-3388/`.

Croissant metadata. A Croissant metadata record documenting the dataset accompanies the full release.

Benchmark code. During review the code for data preparation, training, evaluation and the results tables is available from an anonymised repository `https://anonymous.4open.science/r/BLISS-60DF/`. It is released under the MIT licence. Versions are immutable. The splits are frozen at version 1.0, and any correction is issued as a new version with an erratum. The repository also holds the three split files, and a script in it (`benchmark/data/download.py`) downloads the review sample from the link above, checks its SHA-256 and prepares it for the scripts.

C AUTHOR STATEMENT AND DATA LICENCE CONFIRMATION

We, the authors, bear full responsibility for any violation of rights arising from the use of the data. All simulations in BLISS were performed by the authors. The dataset is released under the Creative Commons Attribution 4.0 International licence (CC BY 4.0), which allows others to share and adapt the work for any purpose, provided that appropriate credit is given. The benchmark code is released under the MIT licence.

Third-party assets. The experimental measurements and micro-CT scans used for validation (App. E.6.3) come from a published study (Derveni et al., 2023), which we cite, and they are not redistributed. The simulations use Abaqus (Dassault Systèmes Simulia Corp., 2024) under a university research licence. The models use PyTorch (BSD-3-Clause), NumPy and SciPy (BSD-3-Clause), PhysicsNeMo (Apache-2.0) and torch-harmonics (BSD-3-Clause). These licences have been respected. Transolver and DeepONet are reimplemented from their papers in the released code.

D BROADER IMPACT

Thin shells carry the loads of pressure vessels, storage tanks, domes, silos and launch-vehicle structures with little material. Their collapse is very sensitive to small imperfections, so engineers still design them with conservative knockdown factors and verify them with expensive tests. To our knowledge, BLISS is the first public dataset that pairs the collapse load, the displacement field at collapse and the failure site of one nonlinear instability, with a split that crosses a change of physical regime, a measure of how close competing outcomes are and a simulation chain checked against experiments. Its impact spans several communities.

1. **Structural design and inspection.** A surrogate that predicts the collapse load of a measured shell from its scan could support less conservative designs, faster screening of manufactured parts and fewer physical tests. The benchmark gives the evidence such a surrogate needs before it is trusted, including the error on dense defect patterns and on deep defects, which a random split hides.
2. **Machine learning near instabilities.** The benchmark separates a smooth output (the field) from a decision (the failure site) that comes from the same solution. It is a testbed for models that must choose between competing outcomes, such as mixture, ranking, generative and uncertainty-aware models, and it scores them on that choice.
3. **Evaluation under physical shifts.** The splits change a physical property of the shells (the depth and the spacing of the defects) instead of the sampling. Other scientific-ML benchmarks can adopt the same design, with shifts that follow the physics of the problem.
4. **Label precision.** The release carries the solver states one step before and after the peak, a finer mesh study and the agreement of the solver with itself under rotation. A model score can be read against the precision of its own label, which few benchmarks allow.
5. **Foundation for follow-up research.** The load paths, the single-factor sweeps and the single- and two-defect shells support studies of defect interaction, multi-fidelity learning, active learning of new simulations and simulation-to-experiment transfer, beyond the three tasks of the benchmark.

Risks of over-trusting a surrogate. The central result is a warning. In distribution the best models approach the precision of the solver, and a practitioner may be tempted to use them on any shell. From sparse to dense shells every geometry model fails, and a controlled edit that brings two defects together lowers κ by 14% in the solver while the models predict almost no change. Fifty dense examples recover the field, and the choice between two competing failure sites stays close to chance (Section 5). A model with an accurate field can therefore still choose the wrong failure site, and a model trained on one defect pattern can be wrong on another without any warning in its field error. A surrogate should be validated on the population it will be used on, and its failure-site predictions should be read against the site ratio q that BLISS releases. The fields are verified by mesh refinement, and the collapse load is checked against eight physical tests, which differ from the released shells. The dataset is therefore unsuitable for certification or for any safety case without independent validation on the geometry and material in question.

Environmental cost. The released simulations took about 21,600 CPU core-hours, and verification and validation a further 2,400. Training and evaluating the benchmark took about 2,130 GPU-hours of training time, from about 5,540 allocated GPU-hours, and the tabular models 2,900 CPU core-hours (App. A). Releasing the data and every prediction lets others build on these results without repeating this cost.

BLISS contains no personal data and involves no human or animal subjects. Beyond shells, the benchmark is a template for learning problems where one simulation gives both a smooth response and a selection between competing outcomes, such as crack paths, wrinkling patterns or the buckling direction of columns.

E DATASET DETAILS

This appendix describes the shell and its simulation (E.1), the core population (E.2), what a record holds (E.3) and the splits (E.4).

E.1 THE SHELL AND THE SIMULATION

Geometry. Every shell with fields is a hemisphere of outer radius $R = 25.4$ mm and wall $t = 0.23$ mm, clamped at the rim and loaded by uniform external pressure, as in Lee et al. (2016), with the radius, thickness and defect design of Derveni et al. (2023). Its surface carries N inward Gaussian defects. In the angle β from its centre, defect i is

$$w_i(\beta) = -\delta_i \, t \, e^{-(\beta/\beta_0)^2}, \qquad \beta_0 = 3.15^\circ,$$

with depth δ_i in wall thicknesses and centre (ϕ_i, θ_i). The defects add up, and all share one width. Centres lie within 60° of the pole, as in Derveni et al. (2023), which keeps every defect centre about 1.5 buckling wavelengths from the clamped rim.

Material and the knockdown factor. The material is neo-Hookean, with $C_{10} = 0.21$ MPa. The knockdown factor is the collapse pressure over the classical collapse pressure of the perfect sphere,

$$\kappa = p_{\text{peak}}/p_c, \qquad p_c = \frac{2E}{\sqrt{3(1-\nu^2)}}\,\eta^{-2} = 137.7 \text{ Pa},$$

with $E = 1.25$ MPa, the measured modulus of the elastomer (Derveni et al., 2023, Sec. 4), $\nu = 1/2$, and $\eta = 110$, the ratio of the mid-surface radius to the wall. The same p_c normalises every shell with fields.

The simulation. Each shell is solved in Abaqus/Standard (Dassault Systèmes Simulia Corp., 2024) by arc-length continuation (Riks, 1979), which follows the load step by step up to and past its peak. Collapse is the first peak of the load, the first local maximum followed by a drop of at least 10^{-3}. The median simulation has 81 steps (30 to 116). The release keeps the state at the peak and the states one step before and one step after it. These two neighbouring states show how sharply the peak state is defined (App. E.6.2).

Why these choices. The experiments that anchor the pipeline control the volume and record the maximum pressure, which is the same first peak (App. E.6.3). The specimens are elastomer, and the same material law reproduces their collapse pressures.

The mesh. All shells with fields share one base mesh of the sphere, with 76,805 nodes and 76,354 quadrilateral shell elements (S4R, the four-node shell element of Abaqus), stored once (`shared.npz`). A shell's nodes are $\mathbf{p}_i(R + w_i)$, with $\mathbf{p}_i$ the unit direction of node i. On the cap with the defects the elements are about 0.27 mm wide, close to t. Convergence is in App. E.6.1.

Failure site and site ratio. Let $\mathbf{U}$ be the displacement field at the peak. Remove the uniform shrink,

$$\mathbf{D}_i = \mathbf{U}_i - \bar{u}\,\mathbf{p}_i, \qquad \bar{u} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{U}_j \cdot \mathbf{p}_j,$$

where $\bar{u}$ is negative because the shell shrinks inward. The *failure site* is the node of largest $\|\mathbf{D}\|$. A *peak* is a node whose $\|\mathbf{D}\|$ is at least that of its mesh neighbours. The second site is the strongest peak more than 8° from the first, and the *site ratio* $q \in [0, 1]$ is its $\|\mathbf{D}\|$ over that at the first. Shells with $q < 0.7$ are *clear winners* and shells with $q > 0.9$ are *near-ties*. The scorer applies the same rule on its 8,192 nodes (App. G.5). The subsets always use the released q. The sweeps also name the defect nearest the site (App. E.5). Physically, collapse starts as an inward buckle at a defect. The site is where it forms.

E.2 The core population

The core. 4,399 distinct shells, with 2,199 sparse (11 to 18 defects) and 2,200 dense (69 to 81 defects), 196,054 defects in all. κ ranges from 0.357 to 0.668 and the deepest defect from $0.205\,t$ to $9.445\,t$ (Figure 5). The generation stored 4,999 records, of which 600 were the same simulation stored twice. These twins were removed before any split was drawn.

The two spacings. In sparse shells any two defect centres are at least 25° apart. In dense shells at least 10°. These are the two spacings of Derveni et al. (2023). The 25° value is motivated by two-defect shells, whose interaction fades beyond about one buckling wavelength, about 20° on this shell (Hutchinson, 1967; Derveni et al., 2025a). 10° is about half of it. We sample the two values and claim no boundary between them. In the sample, the median distance from a defect to its nearest neighbour is 26.1° in sparse shells and 10.5° in dense ones.

Does the deepest defect decide the site? Table 4 counts, on the 660 B1 test shells, how often the failure site belongs to the deepest defect (the site is assigned to the nearest listed defect). On sparse shells with a clear winner it does on all 94, consistent with the finding that the deepest defect sets the load when defects do not interact (Derveni et al., 2025b; Baizhikova et al., 2026). On dense shells with a clear winner it does on 37.7%, and on near-ties on 20.2% (sparse) and 16.1% (dense), where depth alone does not choose. A random defect would be the deepest one on about 6 to 9% of sparse shells and under 1.5% of dense ones.

Table 4: **How often collapse starts at the deepest defect.** B1 test shells (660), by regime and by the site ratio q of the solved field. Share of shells whose failure site is assigned to the deepest defect (nearest listed defect centre), with the number of shells. The q bands condition on the response, so the rates are conditional. All shells: 40.9%.

	clear winner, $q < 0.7$	0.7 to 0.9	near-tie, $q > 0.9$
sparse, $\beta_{\min} = 25°$	100.0% (94)	47.9% (119)	20.2% (99)
dense, $\beta_{\min} = 10°$	37.7% (114)	26.7% (172)	16.1% (62)

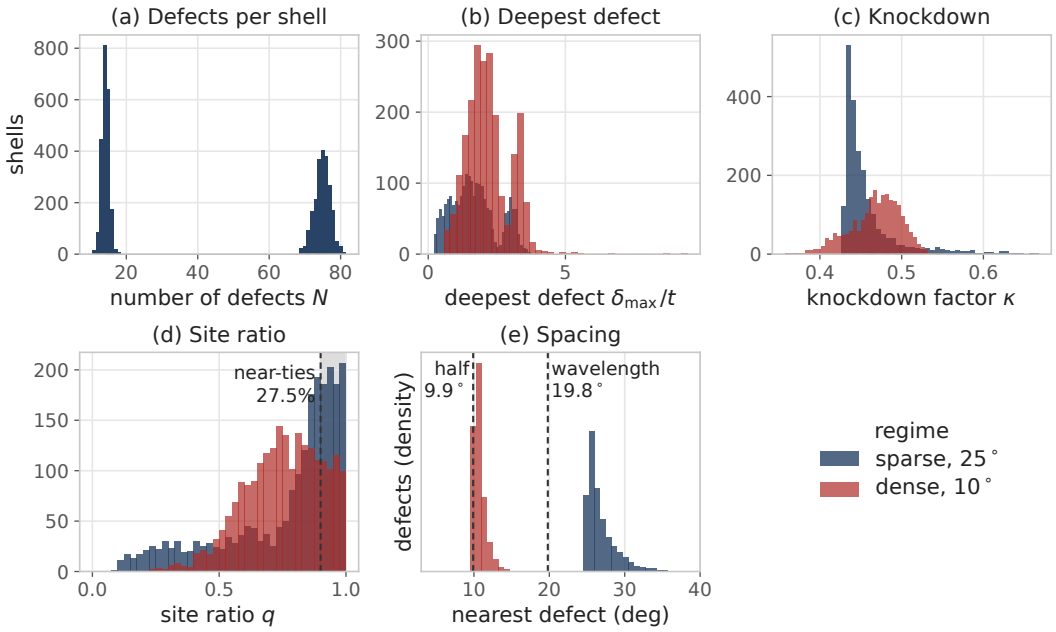


Figure 5: **The core population.** Number of defects, depth of the deepest defect, knockdown factor, the site ratio q (near-ties shaded, 27.5%), and the distance from each defect to its nearest neighbour. Sparse and dense shells in two colours. The dashed lines in (e) mark the buckling wavelength, an arc of 19.8°, and half of it.

E.3 What a record holds in release 1.0

Files. The release holds `shared.npz`, `canonical/` (one record per shell), `metadata/`, `splits/` (`B1.json`, `B2.json`, `B3.json`), the package `bliss/`, `croissant.json` and `hashes/`.

- `shared.npz`. The base sphere `P`, the connectivity `conn`, the radius `R`, a note, and a map to a 180 × 720 grid.
- One record per shell with fields, core and sweeps alike. The input `w`. The targets `U_peak` and `kappa`. The states one step before and after the peak, `U_prev` and `U_next`. `mode`, the change from the peak to three steps after it. The load history `lpf` and the index of the peak `peak_frame`. And three per-shell numbers, the depth ratio `r_d2_d1` of the two deepest defects, the site ratio `q`, and `implied_gap_pct`, a load gap between the two deepest defects estimated from the single-defect curve. About 3.6 MB per shell. The core is about 16 GB.

- Tables, with one row per defect of the core shells (196,054 rows), with the shell's κ. The single- and two-defect shells with their defect parameters and κ, simulated by the authors (no fields or meshes).
- Sweep records in the same format, each with its source id, the value of its edit and the source's split membership. An edit identical to its source is not repeated. It is the core shell itself.
- The split files, the loader, the scorer, Croissant metadata and a SHA-256 manifest (App. B).

The grid view is built on load from the map in `shared.npz` (App. E.6.2).

Table 5: **A release 1.0 record, as a machine-learning object.** Keys of the shared file and of each per-shell record, with their role in the learning problem. Per-node arrays carry the same 76,805 nodes. n_k is the length of the load path (30 to 116). Ranges are over the 4,399 core shells.

key	shape	units	role	range
`shared.npz`, one file				
`P`, `conn`	$(76805, 3)$, $(76354, 4)$	–, index	the domain: base sphere and S4R connectivity, a shell's nodes are $\mathbf{p}_i(R + w_i)$	
`R`, `note`	scalar	mm, –	outer radius, provenance note	25.4
`grid_beta`, `grid_theta`	$(180,)$, $(720,)$	rad	axes of the spherical-grid view	
`knn_idx`, `knn_w`	$(129600, 4)$	index, –	node-to-grid map (four nearest nodes)	
`<id>.npz`, one per shell released with fields				
`w`	$(76805,)$	mm	**input**, and the only one: deviation $\lvert\mathbf{X}\rvert - R$	
`kappa`	scalar	–	**target A**: collapse pressure over the classical one	0.357–0.668
`U_peak`	$(76805, 3)$	mm	**target B**: displacement at the first limit point, its membrane-removed part $\mathbf{D}$ is what the field and site metrics score	
`U_prev`, `U_next`	$(76805, 3)$ each	mm	**label-resolution reference**: the same solve one increment away	
`mode`	$(76805, 3)$	mm	displacement from the limit point to three increments past it	
`lpf`, `peak_frame`	$(n_k,)$, scalar	10^{-5} MPa, index	load path and the limit-point index	
`q`	scalar	–	**evaluation subset**: near-ties $q > 0.9$ (27.5%), clear winners $q < 0.7$	0.098–1.000
`r_d2_d1`, `implied_gap_pct`	scalar each	–, %	depth ratio of the two deepest defects, non-interaction gap estimate	
sweeps only	scalar each	–	source core id, factor value, source split membership	
parametric tables				
δ, ϕ, θ	196,054 rows	t, rad	*not* an input, splits and models on the list of defects	$\delta_{\max}$ 0.205–9.445
$\delta_{\max}$	per shell	t	**split variable** B2 (test above $2.3535\,t$)	0.205–9.445
$\beta_{\min}$	per shell	deg	**split variable** B3 (train 25°, test 10°)	25 or 10

Column dictionary. Every column of every table is described in the release README (App. B).

E.4 SPLITS

The three protocols. Three split files, seed 2027, each a partition of the 4,399 core shells. **B1**, random, 3,299 / 440 / 660. **B2**, deeper defects. The test set is every shell whose deepest defect exceeds $2.3535\,t$. 2,789 / 376 / 1,234. **B3**, sparse to dense. Train and validate on sparse shells, test on dense ones. 1,934 / 265 / 2,200. Table 6 gives the composition of each part.

Leakage. The 600 twins were removed before any split was drawn, so no shell appears in two parts. Every sweep shell is an edit of a core shell, and the sweeps are in no split. P1 to P3 start from 100 B1 test shells, so they must not be used for training when scoring B1. Under B2 and B3 most of their sources are training shells, and P4's 80 sources fall in all parts of every split (App. E.4). Each sweep record carries its source's split membership, so a user can keep only the sources in the test set being scored.

The protocols in full. The three protocols are defined above. The paragraphs below add what each one shifts.

Table 6: **Composition of the three splits.** Each split partitions the 4,399 core shells exactly once (checked). Per part. Number of shells, of which sparse ($\beta_{\min} = 25°$, 11 to 18 defects) and dense (10°, 69 to 81), then medians of the deepest defect $\delta_{\max}$ (in wall thicknesses), of κ and of the site ratio q, and the counts of near-ties ($q > 0.9$) and of shells with a clear winner ($q < 0.7$). B1 is the control. Its three parts match. B2 holds out the deepest defects (deepest defect up to $2.35350\,t$ in train and validation, from $2.35351\,t$ in test). Its test part is 62% dense against 45% in train, so B2 is read per regime. B3 trains and validates on the sparse shells and tests on the dense ones.

split	part	shells	sparse	dense	$\delta_{\max}$	κ	q	near-ties	clear
B1 random	train	3,299	1,667	1,632	1.89	0.456	0.81	936	1,043
	val	440	220	220	1.77	0.456	0.79	113	154
	test	660	312	348	1.85	0.459	0.80	161	208
B2 amplitude	train	2,789	1,533	1,256	1.60	0.465	0.80	697	951
	val	376	199	177	1.58	0.468	0.80	95	123
	test	1,234	467	767	3.07	0.436	0.84	418	331
B3 regime	train	1,934	1,934	0	1.59	0.446	0.86	697	561
	val	265	265	0	1.69	0.443	0.85	85	70
	test	2,200	0	2,200	2.09	0.471	0.76	428	774

Table 7: **Evaluation sets.** The test sets are split by defect spacing (sparse / dense) and by site ratio (clear winners $q < 0.7$ / near-ties $q > 0.9$).

set	train / val / test	what shifts	test sparse / dense	clear / near-tie
B1 within distribution	3,299 / 440 / 660	nothing (random split)	312 / 348	208 / 161
B2 deeper defects	2,789 / 376 / 1,234	deepest defect above $2.35\,t$, with 62% dense in test and 45% in train	467 / 767	331 / 418
B3 sparse to dense	1,934 / 265 / 2,200	sparse in training, dense in test (spacing and number of defects)	0 / 2,200	774 / 428

Edited shells, outside the benchmark. B1 models are applied to the intervention tests P1 to P3, and B3 models to P3 and the dense shells of P4. Each predicted change is compared with the solver's change on the same shell (App. H.10).

B2, deeper defects. Training and validation shells have a deepest defect of at most $2.35\,t$. Test shells have a deeper one. The threshold is stored in the split file. The collapse load of a single defect almost stops falling once the defect is about one wall thickness deep (it still falls about 6% from $1\,t$ to $5\,t$). Every test shell and about four in five training shells are past that point, so B2 extrapolates in the input more than in the label. The test set is also denser, with 62.2% dense shells, against 45.0% in training and 47.1% in validation. We therefore report B2 per regime as well.

B3, sparse to dense. Training and validation shells are sparse, and test shells are dense. The weights and the epoch choice never see a dense shell. The few-shot variant is described in App. G.2.

Subsets. Near-ties are 161 of 660 test shells on B1, 418 of 1,234 on B2 and 428 of 2,200 on B3. Clear winners are 208, 331 and 774. The subset scores are therefore read on different compositions.

Every test shell is scored. The scorer excludes no test shell on any protocol.

The sweeps and the splits. The sweeps are in no split. Their sources do belong to splits (Table 8), and every sweep record carries its source id and that membership.

Table 8: **Split of the sweep sources** (train / validation / test). P1 to P3 share 100 source shells.

sweeps	sources	B1	B2	B3
P1, P2, P3	100	0 / 0 / 100	56 / 7 / 37	84 / 15 / 1
P4	80	65 / 6 / 9	50 / 8 / 22	34 / 6 / 40

E.5 THE SINGLE-FACTOR SWEEPS

Each sweep takes core shells (the *sources*), changes one thing, and simulates again with the release pipeline. The records have the core format and carry their source id, the value of the edit and the source's split membership. The sweeps are in no split and serve evaluation and analysis only

Table 9: **The parts of BLISS.** The release holds 8,528 simulated shells, all simulated by the authors. The 4,399 core and 1,870 sweep shells are released with fields (6,269), and the 2,259 single- and two-defect reference shells are released as parametric rows with their κ. The unshifted arm of P2 and the all-kept arm of P4 reproduce their core source exactly, so the sweep files hold only new shells, each with its own id and its source id. Every shell released with fields has $R = 25.4$ mm, $t = 0.23$ mm and defect width $\beta_0 = 3.15°$. The single-defect shells have $R = 24.85$ mm and $\eta = 108$, and the two-defect shells $R = 25.4$ mm and $\eta = 100$.

part	records (sources)	what changes	κ from	used for
simulated in this release, released with fields				
core, sparse	2,199	11–18 defects, $\beta_{\min} \geq 25°$	this release	every split, B3 train
core, dense	2,200	69–81 defects, $\beta_{\min} \geq 10°$	this release	every split, B3 test
sweep P1	300 (100)	pattern rotated 30, 60, 90°	this release	evaluation only[a]
sweep P2	750 (100)	deepest defect moved ± 1.5 to $\pm 6°$	this release	evaluation only[a]
sweep P3	500 (100)	one pair from 25 to 10°	this release	evaluation only[a]
sweep P4	320 (80)	far defects kept 75 to 0%	this release	evaluation only[b]
parametric rows, no fields				
core defects	196,054	one row per defect of the 4,399 core shells	this release	models on the list of defects
single defect	399	one defect	this release	$\kappa_1(\delta)$, parametric rail
two defects	1,860	two defects, a set separation	this release	pair interaction, parametric rail

[a] Sources: 100 core shells, 99 sparse, all in the B1 test set, so B1 training never sees them (Appendix E.4). P2 is analysed on the 87 sources with the unmoved arm and at least 7 of its 9 arms (Appendix E.5). [b] Sources: 40 per regime. B1 65 / 6 / 9, B2 50 / 8 / 22, B3 34 / 6 / 40 (train / val / test). Appendix E.5.

(App. E.4). P1 to P3 start from the same 100 core shells, 99 of them sparse. P4 starts from 40 sources per regime. An edit identical to its source is not repeated. It is the core shell itself. Figure 6 summarises the four sweeps.

P1. Rotate the whole pattern. The defects of 100 shells are rotated about the pole by 30, 60 and 90° (300 shells). This changes nothing physically, only how the defects lie on the mesh. κ changes by a median of 0.05 to 0.09%. The site rotates with the pattern on 66 to 78% of shells. Each of the 85 misses is a near-tie. So on near-ties the site depends on details as small as the mesh orientation.

P2. Move the deepest defect. The deepest defect is moved along its meridian by ± 1.5 to $\pm 6°$ (750 shells from 100 sources). The analysis uses the 87 sources that have the unmoved arm and at least 7 of the 9 moves. The other moves were not generated for the remaining 13. κ changes smoothly. A median of at most 0.27% per $1.5°$ step. The site moves to another defect on 25% of shells within $\pm 6°$, and on 16% already at $1.5°$. This is no more than rotation alone does (P1), so it shows the limited precision of the site on the scale of a defect's position rather than a physical move of the site.

P3. Bring two defects together. One pair is brought from 25° to 10° in steps (25, 20, 15, 12.5 and 10°), the other defects fixed (500 shells). Down to $12.5°$ the median κ changes by $+0.6$ to $+0.9\%$, though single shells rise by up to about 15%. At 10° κ falls by a median of 14% (91 of 100 below -2%), and the collapse forms between the pair. At 10° the site lies more than 2° from the centre of every listed defect on 97% of shells, against 2 to 19% at the wider spacings. Near-ties fall from 58% at $12.5°$ to 4% at 10°. This is one pair in sparse shells. It says nothing about spacings between the two sampled values of the core.

P4. Remove far defects. From 40 sources per regime, the two candidate sites and every defect within 20° of them are kept, and the far defects are removed in a random order, keeping 75, 50, 25 or 0% of them (320 shells). In sparse shells, removal never lowers κ by more than 0.35% and raises it slightly (median $+0.08$ to $+1.42\%$). When the site changes, it usually goes to the source's own second site. In dense shells, removal lowers κ (median -2.2 to -4.8% at 75 to 25% kept, down to -19.4%), and in 92 of the 98 edits that lower it by more than 1% the collapse moves away from

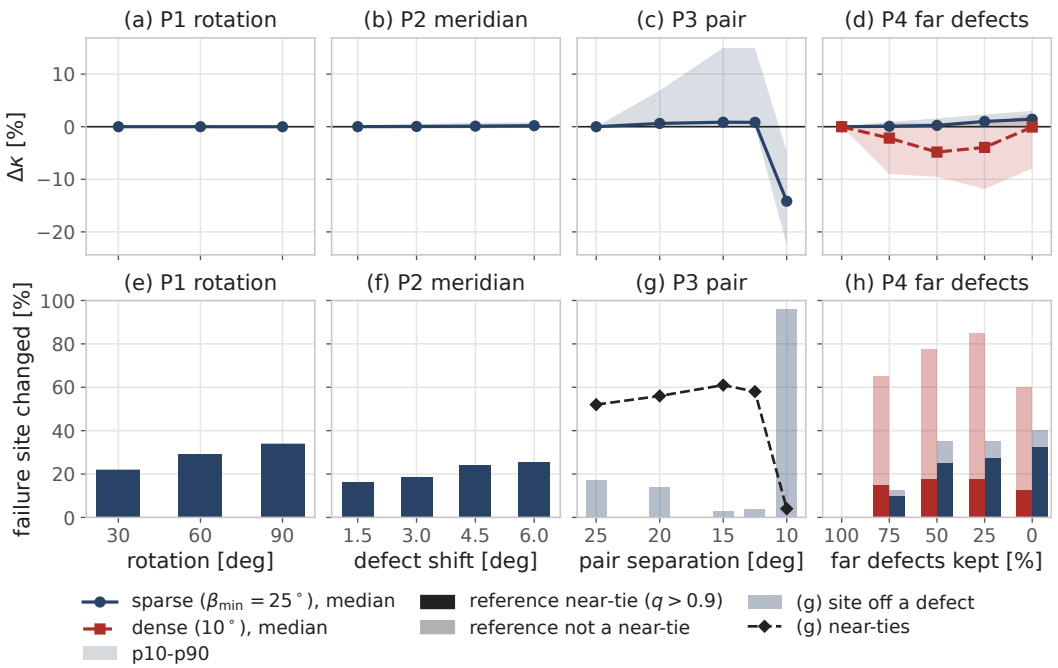


Figure 6: **Small edits move the load little and the site often. In dense shells, removing defects lowers the load.** Top row. Change in κ against the edit (median, band from the 10th to the 90th percentile). Bottom row, share of shells whose failure site changed, dark when the source is a near-tie. (a, e) P1, the whole defect pattern rotated about the pole (100 shells). A change means the site did not rotate with the pattern. (b, f) P2, the deepest defect moved along its meridian (87 of 100 sources). (c, g) P3, one pair brought from 25° to 10° (100 shells). Bars. Sites off every defect. (d, h) P4, defects far from the two candidate sites removed (40 sources per regime), sparse and dense.

both candidate sites, usually onto a far defect whose own shape is unchanged but which lost most of its neighbours. With all far defects removed the median returns to -0.1%. The drop appears at every source depth from $0.5\,t$ up, so the depth of the sources does not explain it. The collapse rule does not cause it. The first peak is the highest one on every run but two, and the highest peak gives the same medians. In dense shells, a defect stripped of its neighbours can become the weakest point.

Reading. Rotating the pattern or moving one defect changes κ little, while the site can flip, above all on near-ties. In sparse shells, removing far defects changes the load little (median at most $+1.4\%$). In dense shells, both the load and the site depend on how the defects are arranged around each other.

E.6 Verification and validation

Verification asks whether the simulation is accurate. Mesh refinement (E.6.1) and the states around the peak (E.6.2). Validation asks whether it matches real shells. Eight scanned specimens, for the collapse load (E.6.3) and the failure site (E.6.4). Table 10 summarises both.

E.6.1 Is the mesh fine enough?

Six core shells (three sparse, three dense) were solved on four meshes made by the same generator, from 17,975 to 305,416 elements. The finest is the release mesh with every element split into four (Figure 7). At the release mesh, κ is within 0.4% of the finest mesh on five shells (0.13 to 0.38%) and 0.93% on the sixth. The failure site moves by at most 0.44°, except on that sixth shell ($q = 0.87$), where it switches to the other local deflection. The displacements at the peak, after removing the uniform shrink, differ from the finest mesh by 0.55 to 1.12% on the sparse shells and 1.2 to 5.7% on the dense ones. The release-level input reproduces the released κ and displacements exactly.

Table 10: **What is checked, and how.** Verification compares the simulation with a finer one. Validation compares it with real shells.

quantity	verification	validation	result	where
κ	four meshes, 6 shells	8 scanned specimens, 10 tests each	within 0.4% of the finest mesh on 5 of 6 shells (0.93% on the sixth), within 3.3% of the experiments, median 1.9%, 7 of 8 within one standard deviation	E.6.1, E.6.3
displacements	four meshes, states one step before and after the peak	none	0.55 to 1.12% (sparse) and 1.2 to 5.7% (dense) from the finest mesh, before/after match R^2 0.993 and 0.966 on B1	E.6.1, E.6.2
failure site	finer mesh on 80 shells, rotation of the defect pattern (P1)	photographs of 8 specimens	moved on 2 of 40 near-ties and 0 of 40 others, near our strongest bulge where one bulge clearly leads	E.6.1, E.6.4
loading before the peak	none	pressure against time, 8 specimens	slope unmatched (-31 to $+28\%$), outside the released quantities	E.6.3

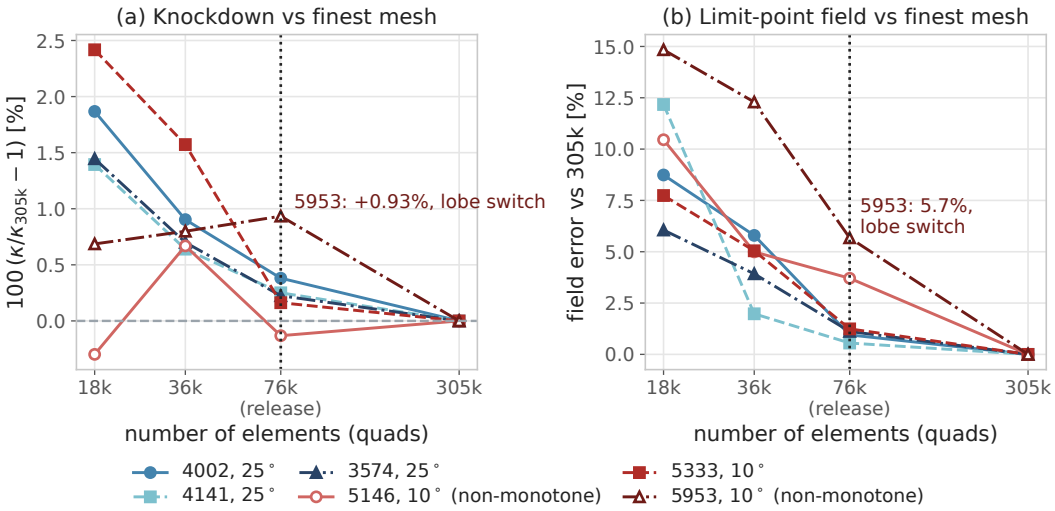


Figure 7: **Mesh convergence at the release mesh.** Six core shells on four meshes, each compared with the finest. (a) the change in κ, in percent. (b) the error of the displacements after removing the uniform shrink. Sparse shells in blues, dense in reds. Open markers where κ does not change monotonically.

Near-ties on a finer mesh. We solved 80 B1 test shells again on the finest mesh, 40 near-ties and 40 shells with a clearer site ($q < 0.6$, since only 16 dense test shells have $q < 0.5$), half of each sparse and half dense, with the design fixed before any solve. The released state served as the coarse side. The site moved by more than 10° on 2 of the 40 near-ties, both to the released runner-up, and on none of the 40 others, a difference without statistical significance ($p = 0.25$). κ changed by a median of 0.26% on near-ties and 0.42% on the others.

E.6.2 How sharply is the peak state defined?

The simulation reaches the peak in steps. Every record keeps the state one step before and one step after the peak. How well these two states match the peak state, scored with the benchmark's field metric, shows how sharply the released displacements are defined. Against the peak state, the two states reach a field R^2 of 0.993 and 0.966 on B1, 0.994 and 0.966 on B2, and 0.990 and 0.931 on B3 (Table 31). On 200 shells the displacements after removing the uniform shrink change by about 10% between consecutive steps. The peak itself does not depend on the drop threshold of the collapse rule (unchanged from 10^{-4} to 3.3×10^{-2} on the 878 generated records with more than one peak).

The grid view loses little. Mesh to grid and back changes w by 1.85% and the displacements by 0.60% (median).

E.6.3 Does the simulation predict the collapse load of real shells?

Table 11: **The eight specimens. Load, path and site.** Measured. Mean and standard deviation of ten collapses. Ours. Our simulation of the scanned specimen at $E = 1.25$ MPa, the measured modulus, with the range from $E = 1.24$ and 1.26 MPa (its ± 0.01 MPa uncertainty) in brackets. Its difference from the measured mean, and that difference in units of the standard deviation (z). Slope. Experimental pre-collapse slope relative to ours (20 to 80% of the peak). Site, at our first limit point. The site ratio q (Appendix E.1) and the great-circle distance from the observed site to our largest and second lobes. Descriptive. The sites are compared without a match rule. As in Derveni et al. (2023), the experimental spread is the standard deviation of the ten tests, and the simulation, which is deterministic, is given with the effect of the modulus uncertainty ($E = 1.25 \pm 0.01$ MPa).

	measured	ours	diff.		slope		distance to lobe [°]	
specimen	[Pa]	[Pa]	[%]	z	[%]	q	largest	second
1	176.2 ± 4.5	179.6 (177.8–180.8)	+2.0	0.76	+23	0.93	59.4	38.8
2	158.8 ± 2.2	156.8 (155.2–158.0)	−1.2	0.88	+19	0.87	21.2	27.5
3	157.9 ± 1.3	153.6 (151.9–154.7)	−2.7	3.39	+26	0.98	81.5	63.8
4	147.4 ± 4.7	146.2 (145.1–146.8)	−0.8	0.24	+9	0.54	10.5	102.3
5	97.3 ± 3.5	96.2 (95.5–96.9)	−1.1	0.31	+14	0.73	66.7	16.6
6	87.7 ± 4.5	86.0 (85.2–86.8)	−1.9	0.37	−5	0.99	65.0	13.6
7	78.8 ± 3.5	77.1 (76.5–77.8)	−2.1	0.47	−31	0.23	6.1	34.8
8	137.5 ± 5.4	133.0 (132.0–134.1)	−3.3	0.83	+28	0.44	16.2	77.0

The specimens. Derveni et al. (2023) made, tested and scanned the specimens. This is the only external input to BLISS. There are eight elastomer hemispheres ($R = 25.4$ mm, wall about 0.30 mm. Specimen 8 about 0.43 mm). All eight appear in that paper, with the same numbering. Each was pressurised to collapse ten times at 0.6 ml/min. The ten peaks scatter by 0.8 to 5.2% of their mean (median 3.4%). The experiments provide a quantitative validation of the finite-element framework used throughout BLISS, which is then used to generate the controlled population of the benchmark (Derveni et al., 2023).

The chain. Each scan gives the outer surface and the wall thickness. We place every node of the release mesh on the scanned surface, give it the scanned thickness, and solve with the release pipeline at $E = 1.25$ MPa (bars from 1.24 and 1.26 MPa). The simulation is not fitted to the tests. As in Derveni et al. (2023), the experimental spread is the standard deviation of the ten tests, and the simulation, which is deterministic, is given with the effect of the modulus uncertainty ($E = 1.25 \pm 0.01$ MPa).

Result. The simulated collapse pressure differs from the measured mean by -3.3 to $+2.0\%$, median 1.9% (Table 11, Figure 8). Seven of the eight lie within one standard deviation of their ten repeats. Specimen 3 differs by -2.7% but has the tightest repeats (0.8%). Derveni et al. (2023) report 0.7 to 3.4% with their own simulations of the same scans. Ours use an independent mesh and pipeline. Before the peak, the experiment is stiffer than the simulation on six specimens and softer on two (Figure 9). We did not identify the cause, and it does not affect the peak or any released quantity.

E.6.4 Does the simulated shell collapse where the real one did?

Derveni et al. (2023) photographed each collapsed specimen and marked where it started to collapse. Figure 10 and Table 11 use their numbering of the specimens.

How the angles are measured. The photographs show the shell from above. We read the marked point on this orthographic top view, where a point at polar angle β lies at a distance $\sin\beta$ from the centre of the disk, and convert it to a polar and an azimuthal angle. The scan of each specimen is registered to the design drawing of its defects by the rotation and mirror that best correlate the two. On specimen 4, whose design has one central defect, this fit is nearly tied between two rotations, and

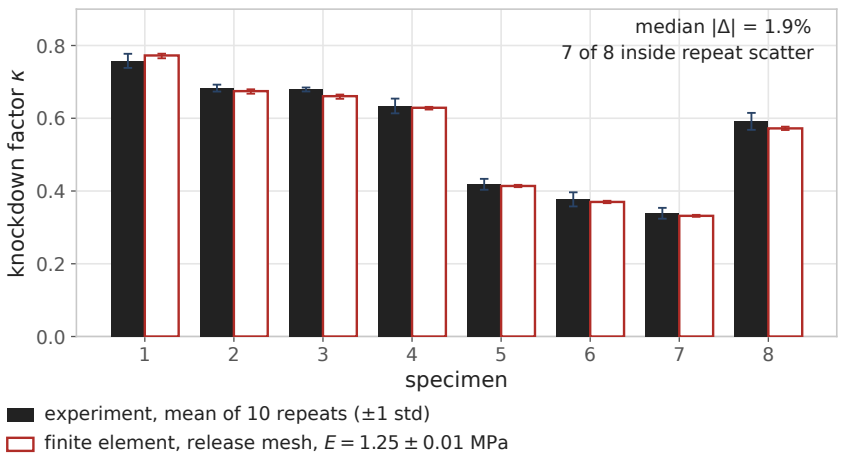


Figure 8: **Collapse load, measured against simulated.** Measured. Mean of ten repeats, ± 1 standard deviation. Simulated. Our simulation of the scanned specimen at $E = 1.25$ MPa, bars from 1.24 and 1.26 MPa. Both as κ on one classical pressure, computed with $t = 0.30$ mm. As in Derveni et al. (2023), the experimental spread is the standard deviation of the ten tests, and the simulation, which is deterministic, is given with the effect of the modulus uncertainty ($E = 1.25 \pm 0.01$ MPa).

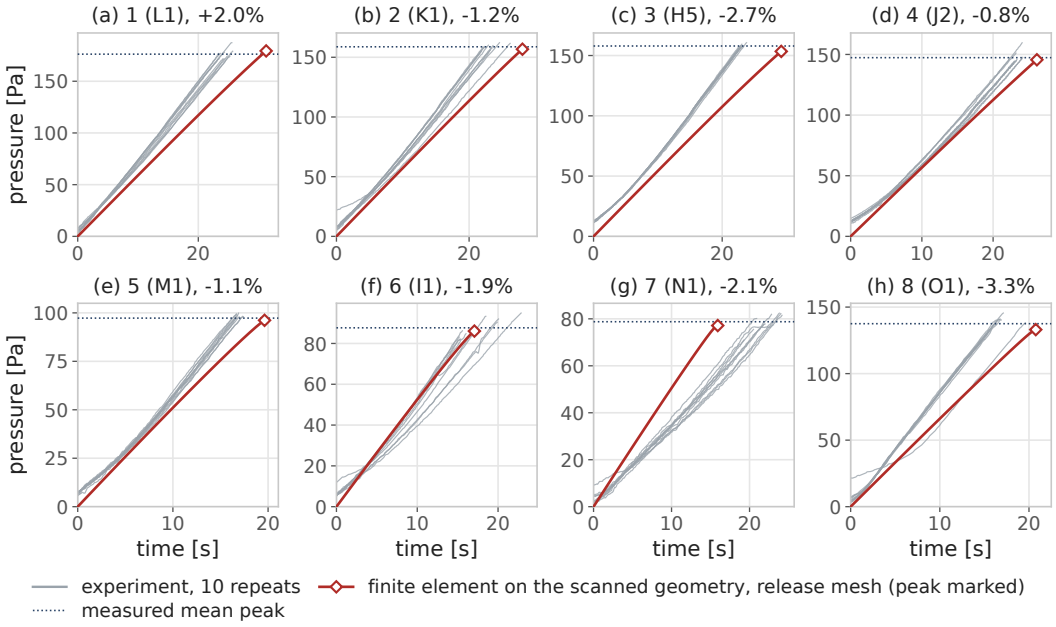


Figure 9: **Pressure against time, eight specimens.** The ten measured repeats and our simulation, peak marked. The simulated time is the volume swept by the deformed shell divided by the pump rate. The simulation is not fitted to the tests.

the ink reference mark on the photographs fixes the orientation. The distance is the great-circle angle between the direction of the marked point and the direction of each local deflection of our simulation just past the peak. The reading is robust near the pole. Near the rim the top view compresses the polar angle, so an error in the image grows by $1/\cos\beta$, a factor of about 1.7 at $55°$. The angular uncertainty of the observed site after registration was not quantified.

Where our simulation has one clear site ($q \leq 0.54$, specimens 4, 7 and 8), the real shell collapsed within $16.2°$ of it. Where two local deflections are close in size, it collapsed near our second one, between the two, or away from both (Figure 11). Derveni et al. (2023) noted that their shells 1 and 3 collapse away from their defects. In our simulations both are near-ties ($q = 0.93$ and 0.98), as is specimen 6, which is consistent with a site that small differences can decide. Eight specimens describe this pattern, and they are too few to test it.

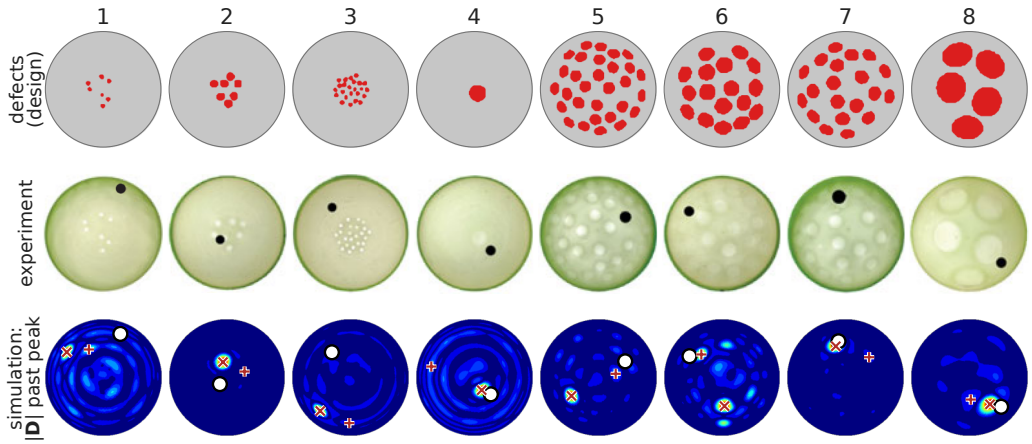


Figure 10: **Where the specimens collapsed, and where our simulation does.** For each specimen, the figure shows the design defects, the photograph after collapse with the observed site marked (Derveni et al., 2023), and the local deflections of our simulation just past the peak, in the same orientation, with the strongest one marked. The photographs are from the experiments of Derveni et al. (2023).

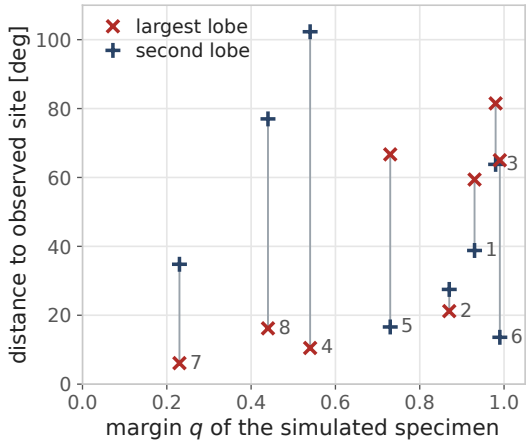


Figure 11: **The observed site against our site ratio.** For each specimen, the distance from the observed site to the strongest and to the second local deflection of our simulation, against its site ratio q.

F PHYSICS BASELINE IN DETAIL

The weakest-link rule is the physics baseline for the load. It needs no training and it reads the same input as the geometry models. It follows the idea that a shell is about as weak as its worst defect (Baizhikova et al., 2026), reduced to its simplest deterministic form. This section builds it in four steps and then describes the other references.

Step 1, the depth of the worst defect. The rule reads the depth of the deepest point of w on the full 76,805-node mesh and divides it by the wall thickness, $\delta_{\max} = \max_i(-w_i)/t$ with $t = 0.23$ mm. The depth read from w is close to the listed depth of the deepest defect (median ratio 0.99).

Step 2, the single-defect curve. A shell with one defect of depth δ collapses at $\kappa_1(\delta)$. We simulated 399 such shells (App. E.3). They have the same wall and defect width as the core and a slightly smaller radius ($R = 24.85$ mm, $R/t = 108$ against 110 for the core). The curve falls steeply up to a depth of about one wall thickness and then slowly, by about 6% between $1\,t$ and $5\,t$ (right panel of Figure 12). It is used as is, with no correction for the small difference in radius.

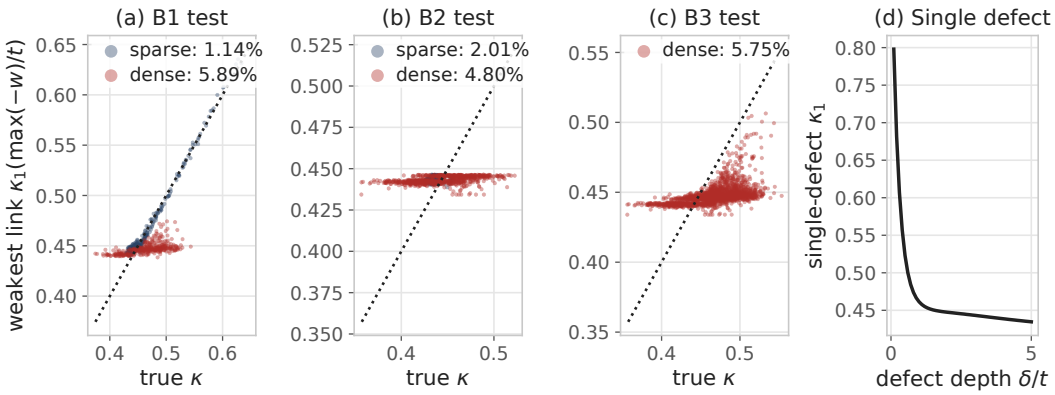


Figure 12: **The weakest-link rule against the true knockdown factor.** Left three panels show the rule $\kappa_1(\max(-w)/t)$ against the true κ on the B1, B2 and B3 test sets, coloured by regime, with the median relative error in the legend. The right panel shows the single-defect curve $\kappa_1(\delta)$ that the rule reads.

Step 3, the prediction. The rule predicts $\hat{\kappa} = \kappa_1(\delta_{\max})$. Depths beyond the end of the curve are clamped to its end. This happens on 0, 7 and 7 test shells of B1, B2 and B3.

Step 4, does the rule match the data? On sparse shells it does well, with a median relative error of 1.14% on B1 and 2.01% on B2. On dense shells the error grows to 5.89% on B1, 4.80% on B2 and 5.75% on B3. The sign of the error explains the difference. On sparse shells the rule is slightly high (median signed error $+1.08\%$ on B1). On dense shells it is low (-4.49% on B1 and -4.66% on B3), so dense shells carry more load than their worst defect alone predicts. Interactions between close defects can raise the buckling pressure as well as lower it (Derveni et al., 2025a), and the rule ignores them by design. Pooled over regimes the error is 2.49%, 2.72% and 5.75% on B1, B2 and B3 (Table 2).

Variants. Reading w on the 8,192 scoring nodes instead of the full mesh gives 2.74%, 2.85% and 5.49%. Reading the depth from the list of defects (a privileged input) gives 2.48%, 2.71% and 5.78%. The rule is therefore limited by the physics it leaves out and hardly by how the depth is read.

F.1 The other reference predictors

Every learned row is read against predictors that need no training or read only w. They are scored by the same function, on the same nodes and shells, as the learned models (App. G.5). Table 12 gives their numbers. Two of them read the list of defects, and are marked. The random-defect null site and the privileged version of the weakest-link rule.

Zero-training references. The *training mean* predicts, for every test shell, the mean field (node by node) and the mean κ of the training shells. The *regime-conditional mean* does the same per regime. On B3 the training set is sparse, so the dense test shells get the sparse mean. Three *null sites* give a floor for the site metrics. A point drawn uniformly by area on the hemisphere, the centre of a randomly chosen defect of the shell, and the site of the training-mean field. The *deepest point of w* predicts the collapse at the scoring node where w is most negative. It is right on 64% of B1 clear winners (a hit is a site within 10°). Section 5.1 quotes a different rule for the same idea. There the site is assigned to its nearest listed defect, and that defect is the deepest on all 94 sparse clear winners. The two rules differ, and the 64% pools sparse and dense shells. On near-ties, the *deeper of the two* candidate sites of the true field (lower w within 5°) is predicted to win, beside chance (50%).

Learned references on w. Three simple learners read the same 8,192-node w as the geometry models. Nearest neighbours find the closest training shells in standardised w and predict the mean of their fields and κ. A single nearest neighbour is reported as the memorisation row. A principal-component projection of w feeds a ridge map to every node and component of the field and to κ.

Gradient boosting on features of w (leading principal components and summary statistics) predicts κ only. Its field is the training mean. All three get 25 trials on B1 validation with the geometry models' selection metric. For gradient boosting, whose field never changes, the choice falls to the validation κ error. They are then held fixed for B2 and B3 (App. G.4). Each is deterministic, so its three seeds are one run.

The solver as a reference. Three solver rows show how sharply the labels are defined. They are not ceilings. (i) The simulation reaches the peak in steps. We release the state one step before and one step after it, and score them as predictions of the released field (Table 31). (ii) A four times finer mesh changes κ by 0.13 to 0.93% on six study shells (App. E.6.1). (iii) Rotating the whole defect pattern about the pole leaves the physics unchanged (before the mesh is applied), and changes the solver's κ by a median of 0.05 to 0.09% (App. E.5). The loss of mapping the true field to the grid and back was not measured on the full test set.

Table 12: **Reference predictors** on the three test sets (B3 pooled). Metrics as in Table 17. The uniform and random-defect null sites as the median angle to the true site over all (shell, draw) pairs, 256 draws per shell, and the training-mean-field site over all shells. The deepest-point rule as the share of clear winners with the predicted site within 10°.

reference	metric	B1	B2	B3
zero training				
training mean	field R^2 of $\mathbf{U}$	0.632	0.558	0.717
	buckling R^2	0.237	0.221	0.340
	κ median rel. error [%]	5.13	8.33	5.11
	near-tie winner rate [%]	50.3	49.0	47.0
contraction	share of the variance of $\mathbf{U}$ [%]	52.9	45.5	60.3
regime-conditional mean	buckling R^2	0.249	0.234	0.340
	κ median rel. error [%]	4.38	7.07	5.11
null site: uniform on the hemisphere	median angle [deg]	64.8	64.8	67.2
null site: random defect centre (list of defects)	median angle [deg]	55.3	55.7	59.1
null site: training-mean field	median angle [deg]	92.4	93.5	92.3
deepest point of w	clear winners within 10° [%]	64	24	43
	all shells within 10° [%]	38	14	29
deeper of the two sites	near-tie winner rate [%]	51.6	47.8	48.4
weakest-link rule (from w)	κ median rel. error [%]	2.49	2.72	5.75
learned on w (25 trials on B1)				
nearest neighbour (tuning chose one)	buckling R^2	0.773	0.806	-0.270
	κ median rel. error [%]	4.79	3.88	4.86
	clear winners within 10° [%]	57	59	21
	near-tie winner rate [%]	54.0	53.8	51.2
PCA and ridge	buckling R^2	0.787	0.537	-11.86
	κ median rel. error [%]	4.26	6.98	31.83
	clear winners within 10° [%]	82	54	31
	near-tie winner rate [%]	55.3	48.3	50.5
gradient boosting on w features	κ median rel. error [%]	1.12	6.90	6.73
solver (label precision)				
solver states one step before / after the peak	buckling R^2	0.993 / 0.966	0.994 / 0.966	0.990 / 0.931
	near-tie winner rate [%]	69 / 80	78 / 82	60 / 70
release mesh vs refined	κ difference [%]	0.13 to 0.93 (six study shells)		
rigid rotation (P1)	median $\lvert\Delta\kappa\rvert$ [%]	0.05 to 0.09 (100 B1-test sources)		
	same collapse defect [%]	72 (215 of 300 rotations, 100 B1 near-ties)		

G BENCHMARK DETAILS

BLISS is a sequence of experiments, each answering one question. Table 14 lists them. The subsections below describe the learned models (G.1), their training (G.2), their input (G.3), how they are selected (G.4), the metrics (G.5) and the bootstrap over shells (G.6).

Table 13: **The experimental programme as seven questions.** Runs count every training run, search trials included. Compute is the training clock (GPU) or the allocated cores (CPU, simulation). Details in Table 14.

question	models	runs	compute
1. Can geometry models learn the map from w?	6 geometry models, 3 simple learners on w	157 + 75 trials, 72 finals, 3 seeds	1,787 GPU-h (with the MLPs)
2. Does κ need the full geometry?	15 tabular families, 3 set models, 2 MLPs on the list of defects	5,194 tabular trials (2,877 reported), 150 refits, 50 MLP trials, 24 finals	2,887 CPU core-h
3. Does a compact physical description explain it?	weakest-link rule, physical- and interaction-feature models, depth rules	feature models within question 2	–
4. Does the input resolution change the results?	full mesh (4), 2,048 nodes (4), coarse grid (1)	ladder and native finals, 3 seeds each	310 GPU-h (native searches in Table 3)
5. Do a few dense examples repair the B3 failure?	3 fine-tuned models, 3 competitors without a network	108 fine-tunings	2.1 GPU-h
6. Did the model learn the physics or a correlation?	the 8 field models on 1,870 edited shells	inference only	0.6 GPU-h
7. Is a failure the model's or the label's?	neighbouring solver states, rotation, finer mesh	80 re-solves	530 core-h

G.1 Learned models

Every field model maps its input to the displacement field at collapse (three components per node) and to κ. The configuration of each model is the winner of its search (Table 23).

DeepONet (Lu et al., 2021). A branch network reads the 8,192 values of w and returns $3p$ coefficients. A trunk network reads the position of a node and returns p basis functions. The field at a node is the product of the two, and κ is read from the branch output. The selected model has $p = 256$ and networks of width 512 and depth 5.

FigConvNet (Choy et al., 2025). The factorised implicit-grid U-Net of PhysicsNeMo on the point cloud of node positions, with w as the point feature and the model's own scalar head for κ. The selected model has four grid levels and a width multiplier of 1.25.

MeshGraphNet (Pfaff et al., 2021). A graph on the nodes, each linked to its k nearest neighbours. Encoders map nodes and edges (relative position and distance) to features, and a stack of message-passing blocks updates them with residual connections. A decoder returns the field and a pooled head returns κ. The selected model has 12 blocks, width 256 and $k = 16$.

SFNO (Bonev et al., 2023). A spherical Fourier neural operator on the 180×720 image of the sphere. The hemisphere is stretched onto the sphere by doubling the polar angle. The input channels are w and the sine and cosine of the polar angle. The output has three displacement channels and one κ channel, whose area-weighted mean is the prediction. The field is sampled back onto the scoring nodes. The selected model has embedding 160 and 6 layers.

Transformer (Vaswani et al., 2017). Each node is a token with its position and w. Pre-norm encoder blocks with full attention over the 8,192 tokens, a per-node head for the field and an attention-pooling head for κ. It stands for plain attention and is not a published surrogate. The selected model has width 384, 8 layers and 8 heads.

Transolver (Wu et al., 2024). The same skeleton with physics attention. Tokens are softly assigned to learned slices, attention runs among the slices, and the result is mapped back to the nodes. The selected model has width 384, 6 layers and 128 slices.

MLP on the list of defects (privileged input). The 20 or 80 deepest defects of the shell, each as its depth and the position of its centre, and the number of defects. Residual blocks of width 768 map them to the full field and to κ. The model sees the list of defects, which a scanned shell does not provide, and it measures how much of the task the list alone solves.

Tabular families (privileged input). Fifteen regressors predict κ from the same list of defects. They are CatBoost, ElasticNet, ExtraTrees, FT-Transformer, a Gaussian process, k-nearest neigh-

Table 14: **Complete experimental matrix.** One row per experimental block, with the question it answers, the models or references, how they are tuned and trained, the splits or variations, and the seeds. B1, B2 and B3 are the three protocols. Sections H and I to K report the results.

block	question	models or references	tuning and training	splits or variations	seeds
geometry to field and κ	can the map from w be learned?	DeepONet, FigConvNet, MeshGraphNet, SFNO, Transformer, Transolver	25 trials of 60 epochs on B1 (27 Transolver, 30 DeepONet), top 2 retrained for 200 epochs	B1, B2, B3 (configuration fixed on B1)	3
MLPs on the list of defects, field and κ	how much does the list of defects carry?	MLP on the 20 or 80 deepest defects	same search and training as the geometry models	B1, B2, B3	3
full native mesh	is 8,192 nodes enough?	FigConvNet, SFNO, Transolver, MeshGraphNet	own search at 76,805 nodes in an earlier, wider space, same selection metric	FigConvNet, SFNO and Transolver B1 to B3, MeshGraphNet B1	3
resolution ladder	does the node count matter?	FigConvNet, MeshGraphNet, Transformer, Transolver at 2,048 nodes, FigConvNet and Transolver at 8,192	B1 configuration	B1	3
SFNO matched grid	does the grid resolution matter?	SFNO-64×128	reported configuration retrained (pure control), and its own search of 25 trials on B1, top 2 retrained for 200 epochs	pure control on B1, search winner on B1, B2, B3	3
tabular families	does κ need the geometry?	15 families (CatBoost, ElasticNet, ExtraTrees, FT-Transformer, GP, kNN, LightGBM, MLP, ResNet, random forest, ridge, histogram GB, SVR, TabNet, XGBoost)	own Optuna search on B1 only, 40 to 225 trials, 25 random start-up, selected on validation median κ error, refit	B1	3
XGBoost across splits	does the B1 configuration transfer?	XGBoost on the list of defects	B1 configuration fixed	B2, B3	3
physical and interaction features	does a compact physical description explain κ?	ExtraTrees on physical features, random forest and XGBoost on interaction features	tuned per split	B1 to B3	3
set models	do defect interactions matter?	Flat, DeepSets, Pairwise	50 trials per split, loss on κ only	B1 to B3, with and without the extra single- and two-defect shells	3
compositional cut	do one- and two-defect shells explain many-defect shells?	set models	trained on single- and two-defect shells only	tested on the core	3
κ or residual target	does the physics baseline help as a target?	set models	target κ or $\kappa - \kappa_{\mathrm{WL}}$	B1 to B3	3
simple learners on w	memorisation or a linear map?	nearest neighbour, PCA and ridge, gradient boosting on w features, regime-conditional mean	25 trials on B1 (same selection metric), fixed for B2 and B3	B1, B2, B3	1
physics baseline	is κ set by the worst defect?	weakest-link rule	none	B1, B2, B3	–
trivial references	what does no learning reach?	training mean, null sites, deepest point of w, deeper of two	none	B1, B2, B3	–
solver resolution	is a failure the model's or the label's?	states one step before and after the peak, P1 rotation, four times finer mesh	re-solves (300 rotations, 80 finer-mesh shells)	all test shells, 100 B1 near-ties, 80 B1 test shells	–
B3 few-shot	coverage or representation?	FigConvNet, SFNO, Transformer, and three competitors without a network	fine-tuning at one tenth of the learning rate, at most 400 steps	$k = 0$, 5, 10, 25, 50 dense shells, five draws per seed for $k \leq 10$	3
intervention tests	physics or correlation?	the eight field models	inference only	B1 models on P1 to P3, B3 models on P3 and dense P4	3
native rescoring	are the tracks comparable?	FigConvNet, SFNO, Transolver, MeshGraphNet (76,805 nodes)	predictions cut to the 8,192 scoring nodes	B1	3

Totals for the field models. 207 trials in the final B1 search (8 models, pruned trials included, Table 23), 48 long retraining runs (2 configurations $\times$ 3 seeds $\times$ 8 models) and 48 runs on B2 and B3, so about 300 field-model trainings in this first layer. Few-shot adds 108 fine-tunings. Tabular fits number 5,194 Optuna trials in 79 studies (2,877 in the 48 studies behind reported rows) and 150 final refits. The GPU-hours in Table 3 also include the earlier, wider searches and the full-mesh searches.

bours, LightGBM, an MLP, a tabular ResNet, a random forest, ridge regression, histogram gradient boosting, support-vector regression, TabNet and XGBoost. ExtraTrees also reads a set of physical features, and a random forest and XGBoost read features of the defect interactions. Both feature lists are defined in `benchmark/parametric/defect_features.py` of the code release.

Set models (privileged input). Three models read the list of defects as a set. The flat model concatenates the defects in order of depth, DeepSets pools a per-defect embedding, and the pairwise model adds an embedding of every pair of defects. They predict either κ or its difference from the weakest-link rule, $\kappa - \kappa_{\mathrm{WL}}$, and they can be trained with the single- and two-defect shells added to the training set.

G.2 TRAINING CONFIGURATION

Field models. The six geometry models and the two field MLPs share one training script and one recipe (Table 15). The loss is $\mathrm{MSE}(\mathbf{U}) + 0.3\,\mathrm{MSE}(\kappa)$ on the 8,192 nodes. SFNO, whose output is defined on the grid, is supervised on 8,192 grid cells per epoch, drawn by area. The input w is standardised with one training mean and standard deviation. The field target is divided by its training standard deviation, and κ is standardised. Each model has its own search space (learning rate and weight decay for all, plus the model sizes). The budget is the same for all, 25 trials of 60 epochs on B1 (27 for Transolver and 30 for DeepONet, because pruned trials count and a few were re-run).

Table 15: **Search constants**, shared by the six geometry models and the two field MLPs on the list of defects.

search	25 trials on B1, 60 epochs each, tree-structured Parzen estimator, 7 start-up trials
pruner	median rule after 7 trials and 30 epochs, checked every 5 epochs, pruned trials count, failed ones are re-run
searched for all	learning rate log-uniform in $[10^{-4}, 2 \times 10^{-3}]$, weight decay log-uniform in $[10^{-5}, 10^{-1}]$
confirmation	top 2 configurations, 200 epochs, seeds 0, 1 and 2, the higher seed mean wins
selection metric	validation field R^2 after removing the uniform shrink, on the region that buckles
optimiser	AdamW, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, warm-up over 5% of steps, cosine decay to $0.01\times$ the learning rate
training	weight average (EMA 0.999) used for evaluation, bf16, gradient clip 1, batch of 8 shells (not searched)

Tabular models. The tabular families are tuned differently from the field models. XGBoost and the other gradient-boosting, kernel, linear and tabular-network families each had their own Optuna search on B1 only, with 40 to 225 trials by family and 25 random start-up trials. They are selected on the B1 validation median κ error and then refitted with three seeds. The XGBoost configuration chosen on B1 is kept for B2 and B3. The set models had 50 trials on each split's own validation set, a loss on κ only, and three seeds.

Few-shot adaptation on B3. A fixed pool of 200 B3 test shells is set aside and the other 2,000 are scored. A B3 model is fine-tuned on $k = 5$, 10, 25 or 50 shells drawn from the pool. Fine-tuning uses AdamW at one tenth of the training learning rate, 5 warm-up steps, cosine decay, a batch of $\min(k, 8)$ shells, 60 passes over the k shells and at most 400 steps. Each model seed is paired with its own draws of the k shells, five draws per seed for $k \leq 10$ and one for larger k. Three competitors see the same k shells and train no network. They predict the mean field and κ of the k shells, the mean of the three nearest of them in w, or the zero-shot prediction shifted by a constant κ and a constant uniform shrink, both fitted on the k shells.

G.3 INPUT CONFIGURATION

Four tracks. The 8,192-node track is read by five geometry models and is the standard track of every reported number. The scoring nodes are drawn once, uniformly and without replacement, by `np.sort(np.random.default_rng(0).choice(76805, 8192, replace=False))`. The 180×720 grid track is read by SFNO. The grid is built from the full mesh, and the output is sampled back onto the scoring nodes. The full-mesh (native) track uses all 76,805 nodes. Its predictions are cut to the same 8,192 nodes before scoring (Table 25). The tabular track reads the list of defects.

Resolution controls. The 2,048-node ladder (FigConvNet, MeshGraphNet, Transformer and Transolver) is scored on its own 2,048 nodes. A 64×128 SFNO control changes input, supervision and output together. It is run as a pure control on B1 and with its own search on B1 to B3 (Table 28).

The list of defects. Each defect is given by its depth in units of t and the polar and azimuthal angles of its centre. The list is sorted by depth. The MLPs keep the 20 or 80 deepest defects. The padding of shorter lists and the encoding of the angles are those of the released training code.

G.4 Model selection on a held-out validation split

The selection metric. Trials, the top-2 choice and every epoch are selected on the validation field R^2 after removing the uniform shrink, on the region that buckles (the nodes where the true $\|\mathbf{D}\|$ exceeds 10% of its maximum). It was fixed before any test set of this version was scored. It differs from the headline field R^2, which uses all nodes. The load plays no part in the selection. The native runs used the same script and selection metric, with their own search at 76,805 nodes in an earlier, wider search space.

Top two, three seeds. The two best configurations of the search are trained for 200 epochs with seeds 0, 1 and 2. The one with the higher mean validation score is kept for all three protocols (Table 23).

B2 and B3. Hyperparameters are chosen once, on B1 validation, which holds both regimes and deep defects. B2 and B3 retrain that configuration with three seeds and choose only the epoch on their own validation sets. The shifts therefore test the trained weights under a configuration chosen once. Each test set is scored once, with the weights of the selected epoch.

Test-set exposure. In this version each test set was scored once per final model, and no model was selected on a test set. A pre-release iteration of the benchmark had scored models, including every configuration of its search, on test sets that overlap the current ones. All 1,234 B2 and all 2,200 B3 test shells were in the earlier test set of the same protocol. 433 of the 660 B1 test shells were in at least one earlier test set (85 in the earlier random one). Some design choices (the splits, the metrics, the search space) were made after that iteration. The selection rule and metric of this version were fixed before any of its test scores was computed.

How to submit a result.

Predictions. One NumPy archive per protocol and seed, with three keys, `ids`, `pred_U` (shells $\times$ nodes $\times$ 3) and `pred_K` (one κ per shell). Give them on the 8,192 scoring nodes and on every test shell of the split. Grid models sample their output onto the nodes. The scorer loads the true fields, κ, the node positions and q from the release by shell id. It rejects a file whose ids differ from the split's test set, whose arrays have the wrong shape or whose values are not finite.
Scoring. `score_submission(pred, split="B1")` returns every headline metric with a bootstrap interval over shells. `score_by_regime` adds the sparse, dense and q-band parts. `compare_seeds` gives the paired difference with any other row.
Report. All three protocols. Pooled and per regime for B1 and B2, pooled for B3. Three training seeds (mean and range). The scorer version and its SHA-256, as listed in the release.
Rules. Choose hyperparameters and architecture on B1 validation only. Choose epochs on each split's own validation set. Score each test set once per final model. Do not train on a sweep shell whose source is in the test set being scored (Table 8).

The leaderboard is a static, versioned table in the release. Each row records the scorer version that produced it.

G.5 Metric definitions

Table 16 lists the four headline metrics and the references each is read against.

All headline and metric-card numbers come from one function, `score()`. It reads one prediction file per protocol and seed, on the 8,192 scoring nodes. Its output keys are listed in Table 17. The reference rows, the secondary metrics and the intervention statistics come from released scripts (App. F.1 and H.10). In our notation, shell s has true and predicted displacements $\mathbf{U}_{s,i}$ and $\hat{\mathbf{U}}_{s,i}$ at node i. The unit vector $\mathbf{p}_i$ points from the centre of the sphere to node i (the outward radial direction). There are $N = 8{,}192$ nodes and S test shells.

Removing the uniform shrink. The mean radial motion of shell s is $\bar{u}_s = \frac{1}{N}\sum_i \mathbf{U}_{s,i} \cdot \mathbf{p}_i$. It is negative, because the shell shrinks inward. The rest of the field is $\mathbf{D}_{s,i} = \mathbf{U}_{s,i} - \bar{u}_s\,\mathbf{p}_i$. The local deformation that locates the collapse. The prediction is treated the same way, with its own $\hat{\bar{u}}_s$.

Table 16: **Headline metrics and the references each must beat.** Every headline is also reported for sparse and dense shells on B1 and B2 (B3 is all dense). Numbers are in Section 5 and App. H.

output	metric	subset	better	read against
knockdown factor κ	median relative error [%]	all test shells	lower	the weakest-link rule and the best tabular model
field	buckling R^2, the R^2 after removing the uniform shrink	all nodes, all shells	higher	the training-mean field
failure site	clear-site accuracy, the share within 10° of the true failure site (median angle also reported)	clear winners	higher	the deepest point of w
near-tie	near-tie winner rate, the share with a larger predicted field at the true failure site than at the second candidate site (true sites given)	near-ties	higher	chance (50%), the deeper of the two sites, and the agreement of the solver with itself under rotation (72% on B1 near-ties)

Collapse load. The headline is the median relative error, $\mathrm{median}_s\,|\hat{\kappa}_s - \kappa_s|/\kappa_s$, in percent. R^2_κ is also printed. It is left unranked on B2 and B3, because κ varies little inside those test sets, so its R^2 is unstable.

Field. Every R^2 is pooled. All shells, nodes and components enter one sum. The reference is the mean of all true values of the evaluated set, $R^2(\mathbf{A}, \hat{\mathbf{A}}) = 1 - \sum(\mathbf{A} - \hat{\mathbf{A}})^2 / \sum(\mathbf{A} - \bar{A})^2$. The headline is $R^2(\mathbf{D}, \hat{\mathbf{D}})$ over all nodes and shells. The selection metric is the same on the region that buckles only, the nodes where $\|\mathbf{D}_{s,i}\|$ exceeds 10% of the shell's maximum (App. G.4). The R^2 of the raw field $\mathbf{U}$ is also printed. The shrink is removed because the load alone fixes it, and it carries 45.5 to 60.3% of the variance of $\mathbf{U}$. Because the reference is the pooled mean, the training-mean field already scores above zero on B1, with 0.632 on the raw field and 0.237 after removing the shrink. The floor is therefore 0.237 instead of 0 (Table 12).

Site. The site of a field is the node with the largest $\|\mathbf{D}_{s,i}\|$. The site error ℓ_s is the angle between the predicted and the true site. The scorer prints its median and the share below 10°, on all shells and on the clear winners. The share on clear winners is the headline. The two candidate sites of a field are its strongest peak and the strongest peak more than 8° away. A peak is a node whose $\|\mathbf{D}\|$ is at least that of its 12 nearest scoring nodes. The site ratio is $q = \|\mathbf{D}\|$ at the second site over $\|\mathbf{D}\|$ at the first ($q = 0$ if there is no second site).

Subsets. Clear winners ($q < 0.7$) and near-ties ($q > 0.9$) use the released q of the true field, computed on the full mesh (App. E.1). So every model is scored on the same shells. The scorer's output files call the two subsets dominant and ambiguous. Shells with $0.7 \le q \le 0.9$ enter only the load, field and all-shell site metrics.

Near-ties. Take the two candidate sites of the true field, the winner t_1 and the runner-up t_2. The model is right if $\|\hat{\mathbf{D}}_{t_1}\| > \|\hat{\mathbf{D}}_{t_2}\|$. The headline, the near-tie winner rate, is the share of right answers. Chance is 50%. The model does not have to find the sites itself. Three more numbers are printed. The ambiguity error is $\mathrm{median}\,|\hat{q} - q|$. Because $\hat{q}$ is computed on the scoring nodes and q on the full mesh, a perfect field does not score zero (0.0069, 0.0048 and 0.0112 on B1, B2 and B3). The same-winner rate is the share of near-ties whose predicted site is within 10° of the true one. The peak-elsewhere rate is the share whose two predicted sites are both at least 10° from both true ones.

Secondary metrics. For the load. The 90th percentile and the mean of the relative error, and the share of shells with $\hat{\kappa}_s > 1.01\,\kappa_s$. Over-predicting κ overstates the load the shell can carry, the unsafe side for design. For the field, for comparison with other benchmarks. The root-mean-square error of $\mathbf{U}$ and of $\mathbf{D}$ in mm ($t = 0.23$ mm), and the mean and median over shells of the relative error $\|\hat{\mathbf{A}}_s - \mathbf{A}_s\|/\|\mathbf{A}_s\|$, for $\mathbf{A} = \mathbf{U}$ and $\mathbf{A} = \mathbf{D}$. The error of $\mathbf{U}$ is dominated by the uniform shrink, which κ fixes, so it counts the load twice. The error of $\mathbf{D}$ is the comparable number. They are printed in the full results (App. H).

Per regime. `score_by_regime` scores the pooled test set, its sparse and dense parts, and the q bands. Each part is scored as its own set, with its own mean as the R^2 reference. B1 and B2 are reported pooled, sparse and dense. B3 is all dense and reported pooled.

Metrics left out. A per-shell average of R^2 (dominated by shells with small fields) and an error normalised by the variance of $\mathbf{U}$ (dominated by the uniform shrink) are not headline metrics.

Table 17: **Metric card.** Every metric the leaderboard reports, what it measures, unit, direction, the subset and its size on the B1 / B2 / B3 test sets, and its role. Headline metrics in bold.

key	measures	unit	better	subset	n (B1 / B2 / B3)	role
kappa_rel_err	median relative error of κ	%	lower	all	660 / 1,234 / 2,200	headline
kappa_r2	R^2 of κ	–	higher	all	660 / 1,234 / 2,200	secondary, unranked on B2, B3 (little spread)
buckling_r2	pooled R^2 of $\mathbf{D}$ (field after removing the uniform shrink)	–	higher	all nodes, all shells	660 / 1,234 / 2,200	headline
buckling_dimple_r2	the same on the region that buckles in the true field	–	higher	region that buckles	660 / 1,234 / 2,200	selection
field_r2, dimple_r2	pooled R^2 of $\mathbf{U}$	–	higher	all nodes / region that buckles	660 / 1,234 / 2,200	secondary
loc_deg_median, loc_within_10	site error, all shells	deg, share	lower, higher	all	660 / 1,234 / 2,200	secondary
loc_dominant_within_10	share of sites within 10°	share	higher	clear winners	208 / 331 / 774	headline
loc_dominant_median	median site error	deg	lower	clear winners	208 / 331 / 774	beside headline
two_way_winner	winner between the two true sites	share	higher	near-ties	161 / 418 / 428	headline
ambiguity_err	median $\lvert\hat{q} - q\rvert$ (floor 0.007 / 0.005 / 0.011)	–	lower	near-ties	161 / 418 / 428	secondary
same_winner	predicted site within 10° of the true one	share	higher	near-ties	161 / 418 / 428	diagnostic
peak_elsewhere	predicted sites at neither true site	share	lower	near-ties	161 / 418 / 428	diagnostic

G.6 Why a bootstrap over shells?

Method. Every interval is a 95% percentile bootstrap over test shells with 2,000 resamples. The test shell is the unit of the benchmark, so a resample draws shells with replacement and recomputes the metric on them. For an R^2 the reference stays at its full-set value, so the interval describes the prediction. The interval covers the finite test set. Training noise is described separately by the three seeds (Table 22).

Paired comparison. Two rows are compared with `compare_seeds`. Each resample draws one set of shells. The metric of each row is computed per seed on it and averaged over that row's seeds, and the difference is recorded. We report the interval of the difference and the share of resamples in which one row does not beat the other, a bootstrap tail fraction that is not a p-value.

Interpretation. A difference is claimed when its interval excludes zero and its sign holds in all three seeds. Each field-model entry is the mean over three seeds with the range. Tabular models are as stated in their rows, and deterministic references have one run. The intervals carry no correction for multiple comparisons, so a claim near the edge of its interval should be read with care.

H Full results

This section gives every number behind Section 5. All rows are scored by the released scorer on the 8,192 scoring nodes, with 2,000 bootstrap resamples over shells. Table 2 in Section 5 summarises the headline metrics of every model and reference on the three protocols.

H.1 Complete tables per protocol

Tables 18 to 20 list every learned model and reference on B1, B2 and B3, pooled and per regime, with the secondary metrics of the metric card (Table 17).

Table 18: **Full results, B1** (all 660 test shells, 8,192 scoring nodes). Learned field models are the mean over three training seeds. The headline columns (κ median error, buckling R^2, clear winners within 10°, near-tie winner rate) carry the mean of the per-seed 95% bootstrap intervals over shells (2,000 resamples), the other columns are 3-seed means. *Unconservative* is the share of shells with $\hat{\kappa} > 1.01\,\kappa$. *w/o shrink* means after removing the uniform shrink. The relative L^2 error $\|\hat{\mathbf{A}}_s - \mathbf{A}_s\|/\|\mathbf{A}_s\|$ is given as its mean and its median over shells, for the full field and without the shrink. Solver rows give the mean only. References and the solver have one run. A dash means the row has no such output. Gradient boosting on w and the weakest link from w predict κ only. The regime columns are 3-seed means of the scorer on the sparse and dense parts. SFNO-64×128 is the matched-grid control of Table 28, scored the same way.

	κ			field							failure site		by regime			
	median rel. err [%]	R^2	unconserv.	buckling R^2	field R^2	defect-region R^2	RMSE [mm]	RMSE w/o shrink	rel. L^2 mean / median	rel. L^2 w/o shrink, mean / median	clear $\leq 10^\circ$	near-tie	sparse κ err	sparse bkl. R^2	dense κ err	dense bkl. R^2
geometry models (8,192 nodes)																
FigConvNet	1.04 [0.92, 1.17]	0.878	0.26	0.985 [0.984, 0.986]	0.992	0.986	0.0051	0.0048	0.080 / 0.074	0.128 / 0.117	0.96 [0.93, 0.98]	0.60 [0.53, 0.68]	0.50	0.992	1.89	0.976
SFNO	0.58 [0.51, 0.66]	0.929	0.17	0.934 [0.921, 0.946]	0.948	0.956	0.0129	0.0100	0.167 / 0.087	0.258 / 0.135	0.98 [0.96, 0.99]	0.65 [0.58, 0.72]	0.27	0.937	1.54	0.931
Transformer	0.65 [0.56, 0.77]	0.909	0.22	0.969 [0.966, 0.972]	0.985	0.973	0.0070	0.0068	0.102 / 0.096	0.168 / 0.171	0.88 [0.83, 0.92]	0.65 [0.58, 0.72]	0.28	0.992	1.59	0.940
Transolver	0.91 [0.78, 1.09]	0.849	0.24	0.922 [0.917, 0.927]	0.963	0.939	0.0110	0.0108	0.167 / 0.160	0.273 / 0.285	0.72 [0.66, 0.78]	0.55 [0.47, 0.62]	0.42	0.972	2.32	0.859
MeshGraphNet	0.70 [0.62, 0.80]	0.916	0.21	0.968 [0.966, 0.970]	0.983	0.971	0.0074	0.0069	0.121 / 0.117	0.196 / 0.193	0.89 [0.85, 0.94]	0.57 [0.49, 0.64]	0.35	0.982	1.65	0.950
DeepONet	1.32 [1.12, 1.55]	0.813	0.31	0.764 [0.751, 0.776]	0.888	0.819	0.0190	0.0189	0.309 / 0.305	0.510 / 0.510	0.62 [0.56, 0.69]	0.54 [0.47, 0.62]	0.68	0.837	2.29	0.671
SFNO-64×128 (matched grid, equivalent search)	0.53 [0.47, 0.62]	0.932	0.18	0.895 [0.870, 0.916]	0.923	0.931	0.0158	0.0126	0.180 / 0.080	0.284 / 0.127	0.98 [0.96, 0.99]	0.65 [0.58, 0.72]	0.25	0.922	1.55	0.860
models on the list of defects (field)																
MLP-20 (20 deepest defects)	1.19 [1.00, 1.35]	0.815	0.27	0.816 [0.796, 0.834]	0.913	0.859	0.0168	0.0166	0.256 / 0.214	0.428 / 0.371	0.71 [0.64, 0.77]	0.54 [0.46, 0.62]	0.58	0.840	2.09	0.786
MLP-80 (80 deepest defects)	1.20 [1.03, 1.39]	0.798	0.28	0.817 [0.799, 0.834]	0.913	0.853	0.0167	0.0166	0.258 / 0.225	0.430 / 0.383	0.72 [0.66, 0.78]	0.55 [0.47, 0.62]	0.52	0.839	2.30	0.789
models on the list of defects (κ only)																
XGBoost (B1 search)	0.72	0.888	0.22	–	–	–	–	–	–	–	–	–	–	–	–	–
flat set model (tuned on B1)	0.87	0.866	–	–	–	–	–	–	–	–	–	–	–	–	–	–
DeepSets set model (tuned on B1)	0.81	0.848	–	–	–	–	–	–	–	–	–	–	–	–	–	–
pairwise set model (tuned on B1)	0.93	0.849	–	–	–	–	–	–	–	–	–	–	–	–	–	–
references																
training mean	5.13	−0.005	0.49	0.237	0.632	0.307	0.0345	0.0339	0.549 / 0.514	0.878 / 0.922	0.00	0.50	5.20	0.121	5.11	0.384
regime-conditional mean	4.38	−0.012	0.49	0.249	0.640	0.316	0.0341	0.0336	0.538 / 0.510	0.859 / 0.907	0.06	0.53	4.30	0.131	4.47	0.398
nearest neighbour ($k = 1$)	4.79	−0.805	0.66	0.773	0.885	0.793	0.0193	0.0185	0.314 / 0.296	0.504 / 0.485	0.57	0.54	2.59	0.793	5.77	0.746
k nearest neighbours (tuned: k=1, weights=distance)	4.79	−0.805	0.66	0.773	0.885	0.793	0.0193	0.0185	0.314 / 0.296	0.504 / 0.485	0.57	0.54	2.59	0.793	5.77	0.746
PCA and ridge (tuned: m=688, alpha=0.0687)	4.26	0.184	0.49	0.787	0.896	0.827	0.0183	0.0179	0.271 / 0.252	0.435 / 0.407	0.82	0.55	4.41	0.869	4.02	0.682
gradient boosting on w (tuned: m=50, lr=0.0268, leaves=67, l2=0.0102)	1.12	0.826	0.27	–	–	–	–	–	–	–	–	–	0.50	–	2.50	–
weakest link from w	2.49	0.511	0.36	–	–	–	–	–	–	–	–	–	1.14	–	5.89	–
solver (label precision)																
solver, increment $k - 1$	–	–	–	0.993	0.996	0.992	–	–	–	–	1.00	0.69	–	–	–	–
solver, increment $k + 1$	–	–	–	0.966	0.965	0.965	–	–	–	–	1.00	0.80	–	–	–	–

Table 19: **Full results, B2** (all 1,234 test shells, 8,192 scoring nodes). Learned field models are the mean over three training seeds. The headline columns (κ median error, buckling R^2, clear winners within 10°, near-tie winner rate) carry the mean of the per-seed 95% bootstrap intervals over shells (2,000 resamples), the other columns are 3-seed means. *Unconservative* is the share of shells with $\hat{\kappa} > 1.01\,\kappa$. *w/o shrink* means after removing the uniform shrink. The relative L^2 error $\|\hat{\mathbf{A}}_s - \mathbf{A}_s\|/\|\mathbf{A}_s\|$ is given as its mean and its median over shells, for the full field and without the shrink. Solver rows give the mean only. References and the solver have one run. A dash means the row has no such output. Gradient boosting on w and the weakest link from w predict κ only. The regime columns are 3-seed means of the scorer on the sparse and dense parts. SFNO-64×128 is the matched-grid control of Table 28, scored the same way.

	κ			field							failure site		by regime			
	median rel. err [%]	R^2	unconserv.	buckling R^2	field R^2	defect-region R^2	RMSE [mm]	RMSE w/o shrink	rel. L^2 mean / median	rel. L^2 w/o shrink, mean / median	clear $\leq 10^\circ$	near-tie	sparse κ err	sparse bkl. R^2	dense κ err	dense bkl. R^2
geometry models (8,192 nodes)																
FigConvNet	1.47 [1.20, 1.80]	0.154	0.44	0.979 [0.978, 0.980]	0.987	0.980	0.0077	0.0073	0.102 / 0.094	0.148 / 0.143	0.78 [0.74, 0.82]	0.55 [0.50, 0.59]	0.31	0.993	4.08	0.959
SFNO	1.29 [1.19, 1.39]	0.514	0.35	0.915 [0.904, 0.924]	0.927	0.929	0.0184	0.0146	0.214 / 0.124	0.276 / 0.172	0.75 [0.71, 0.80]	0.56 [0.51, 0.60]	0.68	0.962	2.85	0.849
Transformer	1.08 [0.98, 1.20]	0.580	0.29	0.957 [0.955, 0.960]	0.976	0.962	0.0106	0.0104	0.140 / 0.145	0.211 / 0.237	0.60 [0.54, 0.65]	0.59 [0.54, 0.63]	0.41	0.991	2.65	0.911
Transolver	1.62 [1.49, 1.81]	0.023	0.31	0.921 [0.917, 0.925]	0.956	0.932	0.0143	0.0141	0.199 / 0.213	0.298 / 0.345	0.31 [0.26, 0.36]	0.54 [0.50, 0.59]	0.66	0.975	3.63	0.847
MeshGraphNet	1.25 [1.15, 1.37]	0.622	0.23	0.937 [0.933, 0.941]	0.957	0.944	0.0142	0.0126	0.180 / 0.136	0.248 / 0.216	0.59 [0.53, 0.64]	0.54 [0.50, 0.59]	0.60	0.985	2.57	0.870
DeepONet	2.38 [2.27, 2.50]	0.446	0.32	0.752 [0.743, 0.761]	0.860	0.811	0.0255	0.0250	0.361 / 0.348	0.531 / 0.532	0.53 [0.47, 0.58]	0.50 [0.46, 0.55]	2.02	0.840	2.94	0.630
SFNO-64×128 (matched grid, equivalent search)	1.75 [1.67, 1.83]	0.511	0.35	0.812 [0.786, 0.835]	0.852	0.845	0.0263	0.0218	0.276 / 0.131	0.375 / 0.195	0.82 [0.78, 0.86]	0.56 [0.51, 0.60]	1.39	0.952	2.80	0.615
models on the list of defects (field)																
MLP-20 (20 deepest defects)	3.14 [2.80, 3.43]	−0.572	0.65	0.345 [0.220, 0.448]	0.597	0.560	0.0417	0.0394	0.475 / 0.419	0.670 / 0.605	0.39 [0.34, 0.44]	0.52 [0.47, 0.57]	1.68	0.263	4.52	0.459
MLP-80 (80 deepest defects)	2.36 [2.17, 2.65]	−0.505	0.70	0.720 [0.709, 0.732]	0.842	0.749	0.0271	0.0265	0.376 / 0.351	0.547 / 0.505	0.46 [0.40, 0.51]	0.50 [0.46, 0.55]	1.26	0.785	5.83	0.629
models on the list of defects (κ only)																
XGBoost (B1 configuration, frozen)	1.69	0.042	0.46	–	–	–	–	–	–	–	–	–	–	–	–	–
flat set model (tuned on B2)	1.37	0.377	–	–	–	–	–	–	–	–	–	–	–	–	–	–
DeepSets set model (tuned on B2)	1.68	−0.353	–	–	–	–	–	–	–	–	–	–	–	–	–	–
pairwise set model (tuned on B2)	2.80	−0.537	–	–	–	–	–	–	–	–	–	–	–	–	–	–
references																
training mean	8.33	−1.786	0.87	0.221	0.558	0.230	0.0454	0.0443	0.608 / 0.554	0.850 / 0.805	0.00	0.49	8.58	0.120	6.89	0.361
regime-conditional mean	7.07	−2.136	0.92	0.234	0.568	0.245	0.0449	0.0439	0.596 / 0.528	0.836 / 0.774	0.00	0.49	6.84	0.109	8.76	0.408
nearest neighbour ($k = 1$)	3.88	−1.230	0.69	0.806	0.888	0.827	0.0229	0.0221	0.305 / 0.226	0.445 / 0.322	0.59	0.54	0.78	0.888	7.61	0.691
k nearest neighbours (tuned: k=1, weights=distance)	3.88	−1.230	0.69	0.806	0.888	0.827	0.0229	0.0221	0.305 / 0.226	0.445 / 0.322	0.59	0.54	0.78	0.888	7.61	0.691
PCA and ridge (tuned: m=688, alpha=0.0687)	6.98	−3.131	0.50	0.537	0.742	0.611	0.0347	0.0341	0.455 / 0.408	0.667 / 0.553	0.54	0.48	6.66	0.755	7.24	0.232
gradient boosting on w (tuned: m=50, lr=0.0268, leaves=67, l2=0.0102)	6.90	−1.082	0.77	–	–	–	–	–	–	–	–	–	8.99	–	4.93	–
weakest link from w	2.72	0.033	0.61	–	–	–	–	–	–	–	–	–	2.01	–	4.80	–
solver (label precision)																
solver, increment $k - 1$	–	–	–	0.994	0.996	0.994	–	–	–	–	0.98	0.78	–	–	–	–
solver, increment $k + 1$	–	–	–	0.966	0.958	0.965	–	–	–	–	1.00	0.82	–	–	–	–

Table 20: **Full results, B3** (all 2,200 test shells, 8,192 scoring nodes). Learned field models are the mean over three training seeds. The headline columns (κ median error, buckling R^2, clear winners within 10°, near-tie winner rate) carry the mean of the per-seed 95% bootstrap intervals over shells (2,000 resamples), the other columns are 3-seed means. *Unconservative* is the share of shells with $\hat{\kappa} > 1.01\,\kappa$. *w/o shrink* means after removing the uniform shrink. The relative L^2 error $\|\hat{\mathbf{A}}_s - \mathbf{A}_s\|/\|\mathbf{A}_s\|$ is given as its mean and its median over shells, for the full field and without the shrink. Solver rows give the mean only. References and the solver have one run. A dash means the row has no such output. Gradient boosting on w and the weakest link from w predict κ only. SFNO-64 $\times$ 128 is the matched-grid control of Table 28, scored the same way.

	κ			field							failure site	
	median rel. err [%]	R^2	unconserv.	buckling R^2	field R^2	defect-region R^2	RMSE [mm]	RMSE w/o shrink	rel. L^2 mean / median	rel. L^2 w/o shrink, mean / median	clear $\leq 10^\circ$	near-tie
geometry models (8,192 nodes)												
FigConvNet	6.68 [6.38, 6.99]	−0.831	0.13	−1.308 [−1.349, −1.266]	−0.370	−0.544	0.0657	0.0534	0.921 / 1.131	1.287 / 1.526	0.31 [0.28, 0.34]	0.48 [0.43, 0.52]
SFNO	9.37 [9.05, 9.67]	−1.883	0.10	−0.587 [−0.614, −0.561]	0.160	−0.310	0.0515	0.0442	0.848 / 0.854	1.201 / 1.214	0.30 [0.27, 0.33]	0.52 [0.47, 0.56]
Transformer	6.45 [6.21, 6.69]	−0.597	0.16	−0.589 [−0.610, −0.568]	0.137	0.011	0.0520	0.0442	0.807 / 0.849	1.186 / 1.221	0.32 [0.28, 0.35]	0.48 [0.44, 0.53]
Transolver	6.80 [6.55, 7.07]	−0.754	0.15	−1.251 [−1.282, −1.221]	−0.258	−0.432	0.0626	0.0525	0.957 / 1.066	1.382 / 1.470	0.31 [0.28, 0.35]	0.48 [0.43, 0.52]
MeshGraphNet	5.99 [5.72, 6.29]	−0.470	0.12	−1.105 [−1.148, −1.061]	−0.282	−0.504	0.0636	0.0510	0.873 / 1.023	1.191 / 1.422	0.32 [0.28, 0.35]	0.51 [0.46, 0.56]
DeepONet	4.72 [4.53, 4.93]	0.077	0.31	0.295 [0.290, 0.299]	0.636	0.359	0.0339	0.0295	0.545 / 0.560	0.819 / 0.834	0.25 [0.22, 0.28]	0.51 [0.46, 0.56]
SFNO-64 $\times$ 128 (matched grid, equivalent search)	6.16 [5.87, 6.44]	−0.615	0.18	−0.713 [−0.752, −0.677]	0.037	−0.344	0.0551	0.0460	0.888 / 0.887	1.247 / 1.245	0.30 [0.26, 0.33]	0.51 [0.46, 0.56]
models on the list of defects (field)												
MLP-20 (20 deepest defects)	div.	div.	1.00	div.	div.	div.	div.	div.	div. / div.	div. / div.	0.05 [0.03, 0.07]	0.46 [0.42, 0.51]
MLP-80 (80 deepest defects)	div.	div.	1.00	div.	div.	div.	792.7316	670.1662	div. / div.	div. / div.	0.02 [0.01, 0.03]	0.49 [0.44, 0.54]
models on the list of defects (κ only)												
XGBoost (B1 configuration, frozen)	6.50	−0.703	0.13	–	–	–	–	–	–	–	–	–
flat set model (tuned on B3)	8.28	−1.226	–	–	–	–	–	–	–	–	–	–
DeepSets set model (tuned on B3)	9.83	−2.097	–	–	–	–	–	–	–	–	–	–
pairwise set model (tuned on B3)	9.30	−2.370	–	–	–	–	–	–	–	–	–	–
references												
training mean	5.11	−0.096	0.28	0.340	0.717	0.375	0.0299	0.0285	0.501 / 0.498	0.820 / 0.795	0.00	0.47
regime-conditional mean	5.11	−0.096	0.28	0.340	0.717	0.375	0.0299	0.0285	0.501 / 0.498	0.820 / 0.795	0.00	0.47
nearest neighbour ($k = 1$)	4.86	−0.455	0.37	−0.270	0.476	−0.087	0.0407	0.0396	0.633 / 0.625	1.036 / 1.017	0.21	0.51
k nearest neighbours (tuned: k=1, weights=distance)	4.86	−0.455	0.37	−0.270	0.476	−0.087	0.0407	0.0396	0.633 / 0.625	1.036 / 1.017	0.21	0.51
PCA and ridge (tuned: m=688, alpha=0.0687)	31.83	−135.262	0.24	−11.864	−4.237	−8.206	0.1286	0.1260	1.716 / 1.577	2.790 / 2.644	0.31	0.50
gradient boosting on w (tuned: m=50, lr=0.0268, leaves=67, l2=0.0102)	6.73	−0.792	0.13	–	–	–	–	–	–	–	–	–
weakest link from w	5.75	−0.239	0.19	–	–	–	–	–	–	–	–	–
solver (label precision)												
solver, increment $k - 1$	–	–	–	0.990	0.995	0.989	–	–	–	–	0.99	0.60
solver, increment $k + 1$	–	–	–	0.931	0.915	0.930	–	–	–	–	1.00	0.70

Failure site per defect spacing. Table 21 splits the clear-site accuracy of the five best field models on B1 and B2 by defect spacing. The B2 test set holds only 11 sparse clear winners, so its sparse column rests on few shells.

Table 21: **Clear-site accuracy per defect spacing** (%, 8,192 nodes, mean over three seeds). The share of clear winners whose predicted failure site lies within 10° of the true one, on the sparse and the dense test shells of B1 and B2. The last row counts the clear winners in each part.

	B1 within distribution		B2 deeper defects	
	sparse	dense	sparse	dense
FigConvNet	96.8	95.0	100	77.4
MeshGraphNet	99.3	81.3	100	57.4
SFNO	98.6	97.1	100	74.5
Transformer	92.6	84.2	100	58.3
Transolver	92.2	55.8	100	28.6
clear winners	94	114	11	320

H.2 Field reconstructions

Figure 13 shows the true and predicted buckling deviation $|D|$ of every field model on four test shells, each chosen by a fixed rule. On the B1 shell every model places the failure site at the deepest defect, as the solver does. On the dense B2 shell FigConvNet, MeshGraphNet and SFNO place the failure site within 10° of the true one, and the Transformer and Transolver 21° and 98° away. On the first B3 shell the five best models reach a buckling R^2 of -0.42 to 0.70 on the shell and place the failure site 65° to 67° from the true one, at the deepest defect, while DeepONet finds it. On the B3 near-tie all five give the runner-up the larger deflection. Figure 14 shows the same shells for the full-mesh track and the SFNO-64×128 track. On the B1 shell every model of these tracks again finds the true failure site. On the B2 shell FigConvNet (76,805 nodes) and both SFNO tracks find it within 5°, and Transolver (76,805 nodes) misses it by 30°. On the first B3 shell all four models place it 66° to 67° from the true one, and on the B3 near-tie all four give the runner-up the larger deflection.

H.3 Training-seed spread

Table 22 gives the headline metrics of each field model for its three training seeds. On B1 and B2 the seeds agree closely. On B3 the buckling R^2 of some models varies widely between seeds (Transolver from -1.870 to -0.842), so B3 is read as a failure region and its rows are not ranked.

Table 22: **Seed spread.** The four headline metrics of each learned field model for training seeds 0 / 1 / 2 on each protocol (8,192 nodes, pooled test set).

model	protocol	κ err [%]	buckling R^2	clear $\leq 10^\circ$	near-tie
FigConvNet	B1	0.96 / 1.10 / 1.07	0.985 / 0.985 / 0.985	0.96 / 0.96 / 0.96	0.60 / 0.60 / 0.62
FigConvNet	B2	1.53 / 1.46 / 1.43	0.979 / 0.978 / 0.979	0.79 / 0.76 / 0.79	0.55 / 0.54 / 0.55
FigConvNet	B3	6.69 / 6.62 / 6.73	−1.267 / −1.359 / −1.298	0.30 / 0.31 / 0.32	0.48 / 0.47 / 0.47
SFNO	B1	0.57 / 0.60 / 0.57	0.941 / 0.933 / 0.929	0.97 / 0.98 / 0.98	0.64 / 0.66 / 0.66
SFNO	B2	1.43 / 1.28 / 1.15	0.923 / 0.904 / 0.918	0.76 / 0.76 / 0.73	0.55 / 0.55 / 0.57
SFNO	B3	7.33 / 7.16 / 13.64	−0.505 / −0.549 / −0.709	0.30 / 0.28 / 0.30	0.54 / 0.51 / 0.50
Transformer	B1	0.70 / 0.62 / 0.64	0.970 / 0.969 / 0.969	0.90 / 0.85 / 0.89	0.66 / 0.61 / 0.68
Transformer	B2	1.11 / 0.99 / 1.16	0.958 / 0.958 / 0.956	0.63 / 0.57 / 0.59	0.59 / 0.57 / 0.59
Transformer	B3	6.30 / 7.21 / 5.84	−0.860 / −0.454 / −0.453	0.33 / 0.29 / 0.32	0.50 / 0.47 / 0.48
Transolver	B1	0.79 / 0.99 / 0.96	0.921 / 0.920 / 0.926	0.72 / 0.74 / 0.71	0.56 / 0.53 / 0.55
Transolver	B2	1.41 / 2.15 / 1.31	0.920 / 0.920 / 0.924	0.32 / 0.30 / 0.31	0.53 / 0.54 / 0.56
Transolver	B3	6.54 / 7.11 / 6.73	−0.842 / −1.870 / −1.042	0.31 / 0.31 / 0.31	0.48 / 0.46 / 0.49
MeshGraphNet	B1	0.70 / 0.72 / 0.69	0.968 / 0.968 / 0.968	0.88 / 0.92 / 0.88	0.52 / 0.57 / 0.61
MeshGraphNet	B2	1.33 / 1.17 / 1.26	0.944 / 0.936 / 0.932	0.59 / 0.57 / 0.60	0.55 / 0.55 / 0.53
MeshGraphNet	B3	6.03 / 5.98 / 5.94	−1.014 / −1.142 / −1.160	0.33 / 0.31 / 0.31	0.52 / 0.49 / 0.51
DeepONet	B1	1.31 / 1.32 / 1.33	0.765 / 0.762 / 0.764	0.61 / 0.63 / 0.63	0.53 / 0.53 / 0.57
DeepONet	B2	2.29 / 2.39 / 2.46	0.752 / 0.756 / 0.750	0.53 / 0.53 / 0.52	0.52 / 0.50 / 0.49
DeepONet	B3	4.59 / 4.70 / 4.89	0.298 / 0.301 / 0.285	0.24 / 0.26 / 0.24	0.52 / 0.53 / 0.48
MLP-20	B1	1.22 / 1.17 / 1.18	0.815 / 0.813 / 0.820	0.74 / 0.70 / 0.67	0.57 / 0.55 / 0.51
MLP-20	B2	2.95 / 2.43 / 4.05	0.499 / 0.648 / −0.113	0.36 / 0.40 / 0.40	0.56 / 0.49 / 0.53
MLP-20	B3	div.	div.	0.05 / 0.06 / 0.04	0.47 / 0.46 / 0.46
MLP-80	B1	1.15 / 1.17 / 1.27	0.818 / 0.816 / 0.818	0.72 / 0.72 / 0.72	0.54 / 0.55 / 0.55
MLP-80	B2	2.27 / 2.34 / 2.48	0.723 / 0.724 / 0.715	0.43 / 0.48 / 0.47	0.51 / 0.49 / 0.50
MLP-80	B3	div.	div.	0.01 / 0.02 / 0.03	0.49 / 0.49 / 0.49

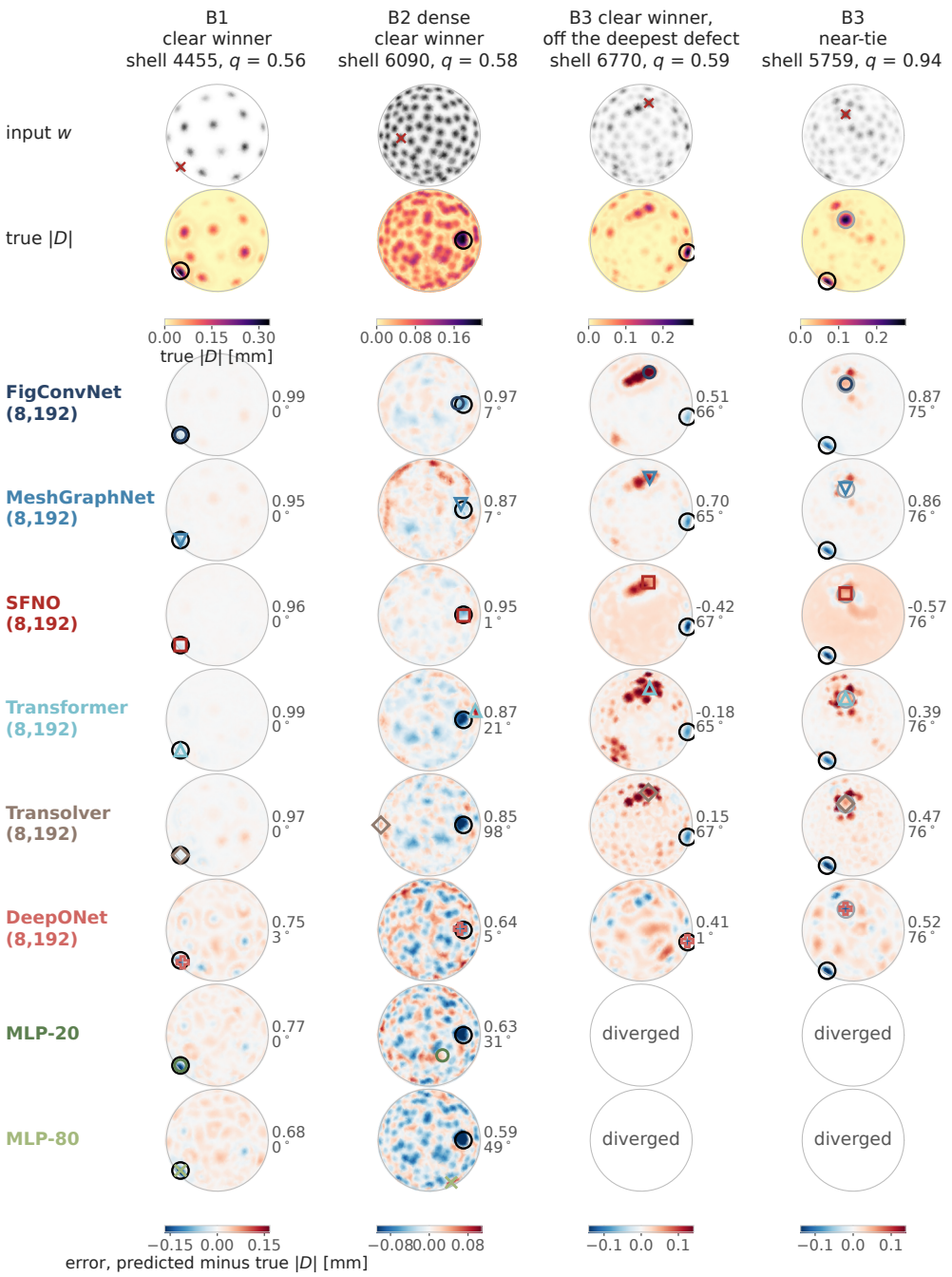


Figure 13: **Field reconstructions on four test shells.** Top view of the cap. Rows show the input w (red cross, deepest defect), the true $|D|$ (deflection without the uniform shrink) and, for each model on 8,192 nodes trained on the protocol of the column (seed 0, Figure 14 for the other resolutions), the error predicted minus true $|D|$. Red is a deflection predicted where there is none, blue a true one that is missed. Black circle, true failure site. Open marker, predicted site. Numbers give the buckling R^2 of the shell and the angle between predicted and true sites. Each shell is picked from a fixed subset, the median one in (1) to (3) and the best-fitted one in (4). (1) B1 clear winners that every model gets right. (2) Dense B2 clear winners. (3) B3 clear winners whose site is away from the deepest defect while the five best models put it there. (4) B3 near-ties where these models pick the runner-up (grey circle). Both MLPs diverge on B3.

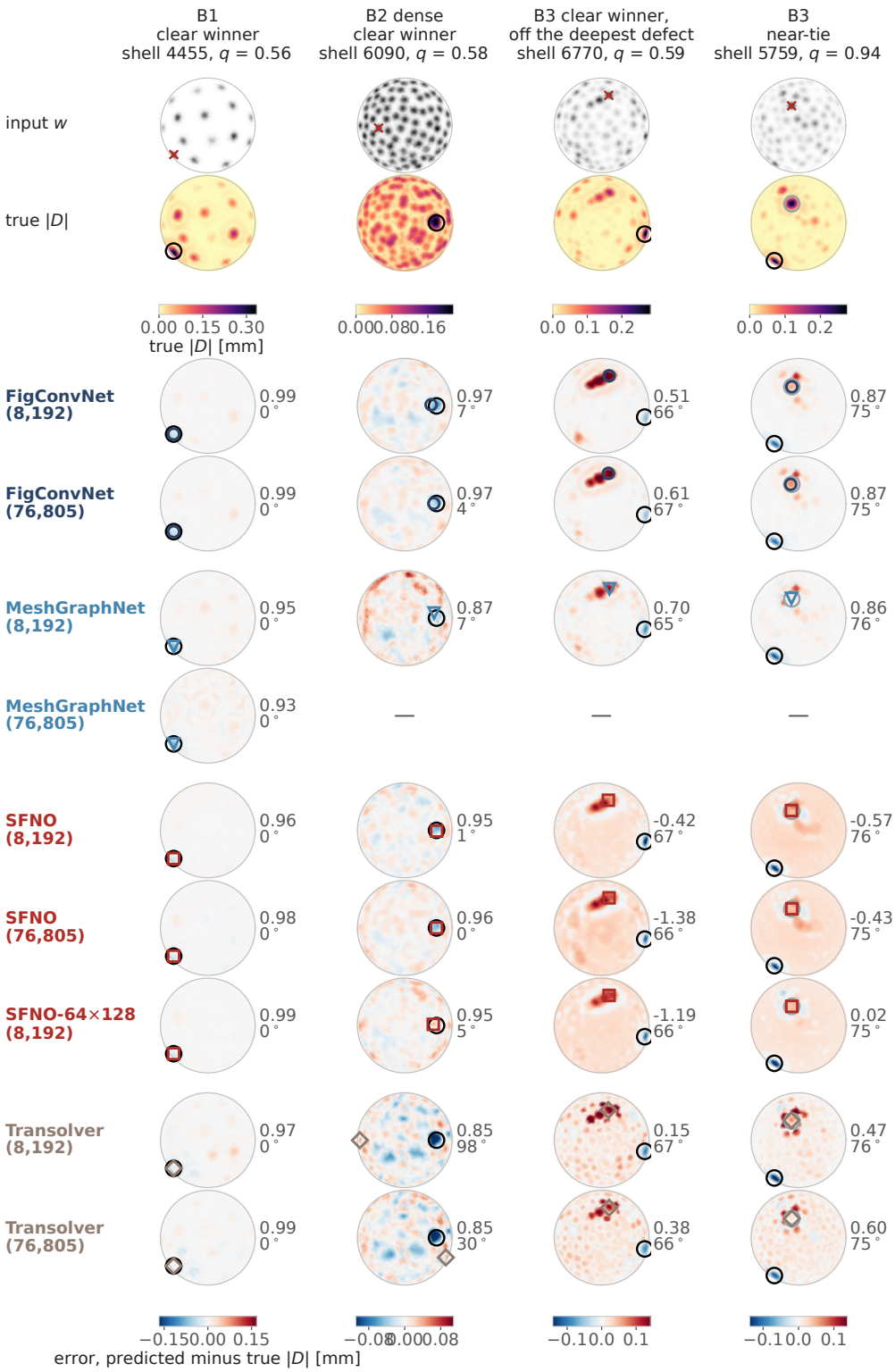


Figure 14: **Field reconstructions at each input resolution.** The shells, colour scales and markers of Figure 13. Each model appears at the resolutions it was trained on, side by side. These are 8,192 nodes, the full mesh of 76,805 nodes (predictions cut to the 8,192 scoring nodes as in Table 25) and, for SFNO, the 64×128 grid of its equivalent search (Table 28). All use seed 0 trained on the protocol of the column. MeshGraphNet appears on the full mesh for B1 only.

H.4 HYPERPARAMETER SEARCH

Table 23 gives the search of each field model, the two finalists and the selected configuration.

Table 23: **Search per model** (B1, TPE with 7 start-up trials, median pruner, 60 epochs per trial), on 8,192 nodes and, below, on the full mesh and on the 64×128 grid of SFNO, each with its own search space. Trials are those of the search budget. Columns give the trials, the completed and pruned trials, the best validation R^2 on the region that buckles, the validation means of the two finalists over three 200-epoch seeds, and the configuration of the winner.

model	trials	compl./pruned	best trial	finalist 1	finalist 2	winner configuration
FigConvNet	25	17 / 8	0.9856	0.9863*	0.9862	lr 0.000115, wd 0.0467, width mult 1.25, levels 4
SFNO	25	11 / 14	0.9402	0.9459	0.9503*	lr 0.000153, wd 0.0107, embed 160, layers 6
Transformer	25	17 / 8	0.9659	0.9754*	0.9750	lr 0.000171, wd 0.00221, dim 384, layers 8, heads 8, dropout 0.0116
Transolver	27	14 / 13	0.9393	0.9435*	0.9217	lr 0.000589, wd 1.77e-05, dim 384, layers 6, slices 128, dropout 0.0198
MeshGraphNet	25	21 / 4	0.9633	0.9722*	0.9722	lr 0.000494, wd 8.68e-05, hidden 256, blocks 12, knn 16
DeepONet	30	20 / 10	0.5659	0.8246*	0.7727	lr 0.000216, wd 0.000613, p 256, width 512, depth 5, dropout 0.00855
MLP-20	25	21 / 4	0.8412	0.8513*	0.8483	lr 0.00119, wd 0.00161, hidden 768, blocks 3, dropout 0.0048
MLP-80	25	18 / 7	0.7882	0.8434*	0.8317	lr 0.000409, wd 0.000451, hidden 768, blocks 3, dropout 0.0563
full mesh, 76,805 nodes, own search in an earlier, wider space						
FigConvNet	27	12 / 15	0.9870	0.9873*	0.9871	lr 0.000169, wd 6.84e-05, width mult 1.25, levels 4
SFNO	25	13 / 12	0.9356	0.9399*	0.9398	lr 0.000378, wd 0.0322, embed 256, layers 6
Transolver	25	14 / 11	0.9424	0.9490*	0.9301	lr 0.000502, wd 0.0245, dim 384, layers 4, slices 64, dropout 0.0589
MeshGraphNet	27	17 / 10	0.9293	0.9401*	0.9296	lr 0.000497, wd 3.79e-05, hidden 128, blocks 12, knn 12
SFNO on a 64×128 *grid, own search*						
SFNO-64×128	25	16 / 9	0.9169	0.9491*	0.9199	lr 0.000543, wd 0.0881, embed 192, layers 8

Figure 15 shows every completed trial of these searches, of the tabular searches of Table 29 and of the searches of the full-mesh and 64×128 tracks (Tables 25 and 28), with the selected configuration marked.

H.5 TRAINING AND VALIDATION CURVES

Figures 16 to 18 show the training and validation curves of the final runs.

H.6 FEW-SHOT ADAPTATION ON B3

Table 24 and Figure 19 give the few-shot curves. Fine-tuning recovers the field of FigConvNet and the Transformer within five dense shells and leaves SFNO below zero. The near-tie winner rate stays between 46 and 52% at every k for every model and competitor. The recalibrated zero-shot model improves only the load.

H.7 FULL NATIVE MESH AND INPUT RESOLUTION

Table 25 scores the native runs on B1 after cutting their predictions to the 8,192 scoring nodes. FigConvNet and Transolver gain a little from the full mesh (buckling R^2 0.987 and 0.928 against 0.985 and 0.922 at 8,192 nodes), and SFNO on the full mesh is below its grid version (0.890 against 0.934). Table 26 gives FigConvNet, SFNO and Transolver (76,805 nodes) on B2 and B3, where they fail on B3 like their 8,192-node versions. Table 27 trains four families on 2,048 nodes. Every family loses accuracy at 2,048 nodes when compared on the same 2,048 nodes, most of all the Transformer and Transolver. Table 28 compares the 180×720 SFNO with a 64×128 grid, as a pure control with three seeds and with its own search on B1 to B3. On B1 the 64×128 grid scores a little lower on the field (buckling R^2 0.880 and 0.895 against 0.934). On B3 it scores below the training mean like the reported SFNO (buckling R^2 -0.713 against 0.340).

H.8 TABULAR MODELS

Table 29 gives the fifteen tabular families on B1 with their number of search trials. Table 30 gives XGBoost with the B1 configuration on every protocol, the best family per split on the list of defects, on physical features and on interaction features, the set models with and without the single- and two-defect shells, the κ target, and the compositional cut.

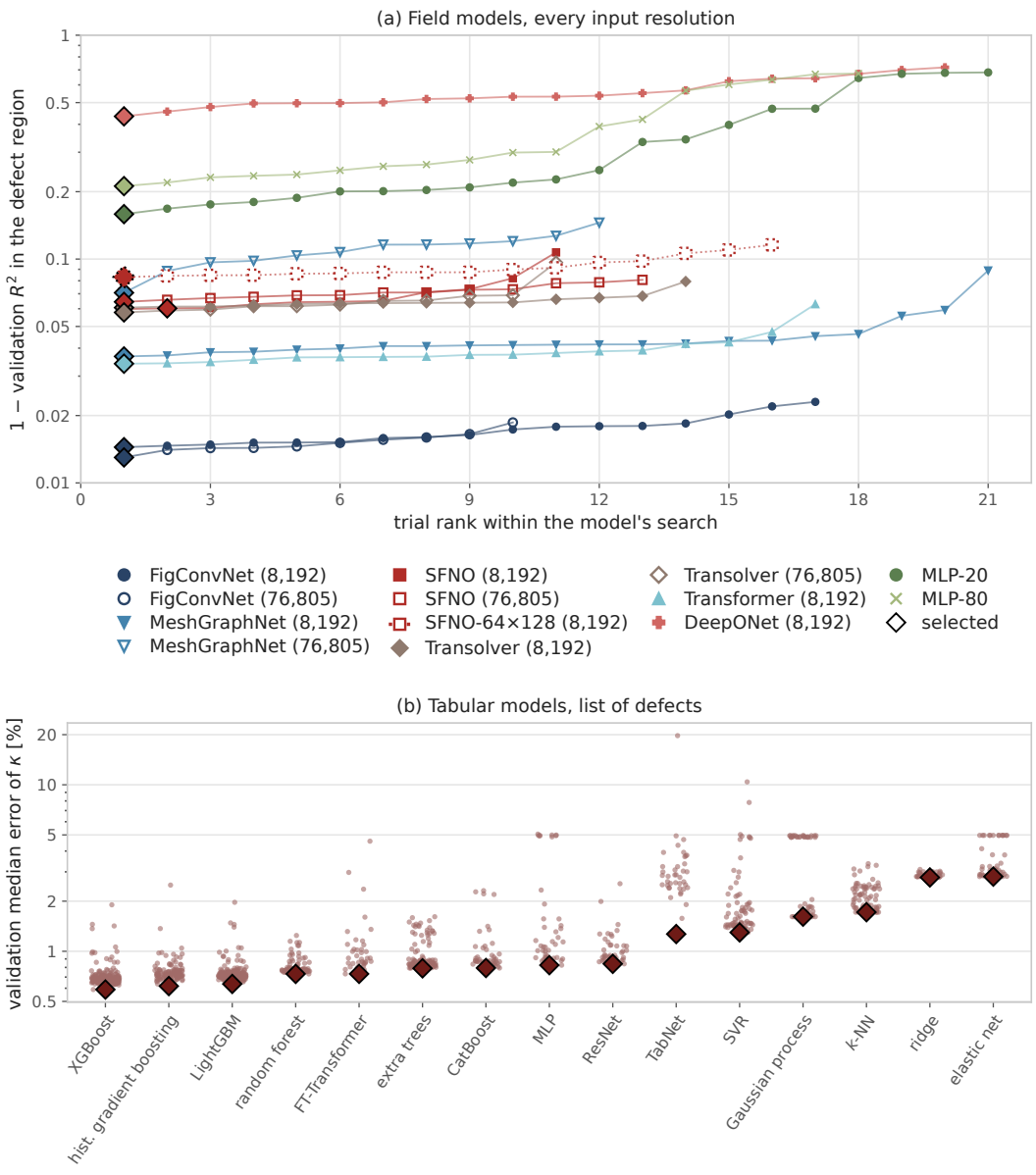


Figure 15: **Hyperparameter search on B1.** (a) Validation buckling R^2 on the region that buckles after 60 epochs, for every completed trial of each field model, plotted as $1-R^2$ against the rank of the trial. Each search appears at its input resolution, 8,192 nodes (filled), the full mesh of 76,805 nodes (open) or, for SFNO, the 64×128 grid (dotted squares), each with its own search space. The MLPs read the list of defects. Only the trials of the search budget are shown, and the diamond is the selected configuration, the better of the two finalists over three 200-epoch seeds. (b) Validation median error of κ for every completed trial of each tabular family, with the selected trial as a diamond.

H.9 Per-shell view of the results

The headline metrics pool the test set. Figures 20 to 23 split them by shell, with the predictions of the three training seeds scored in Table 2. Every number is the mean over the three seeds of a value computed per seed, and the figures give the range over seeds. The per-shell buckling R^2 of a shell compares its error with the test-set mean deflection, so a shell below zero is fitted worse than that mean.

Does B3 fail on every shell? On B1 and B2 almost no shell falls below zero (Figure 20). On B3, 60 to 79% of the shells fall below zero for FigConvNet, MeshGraphNet, the Transformer, Transolver

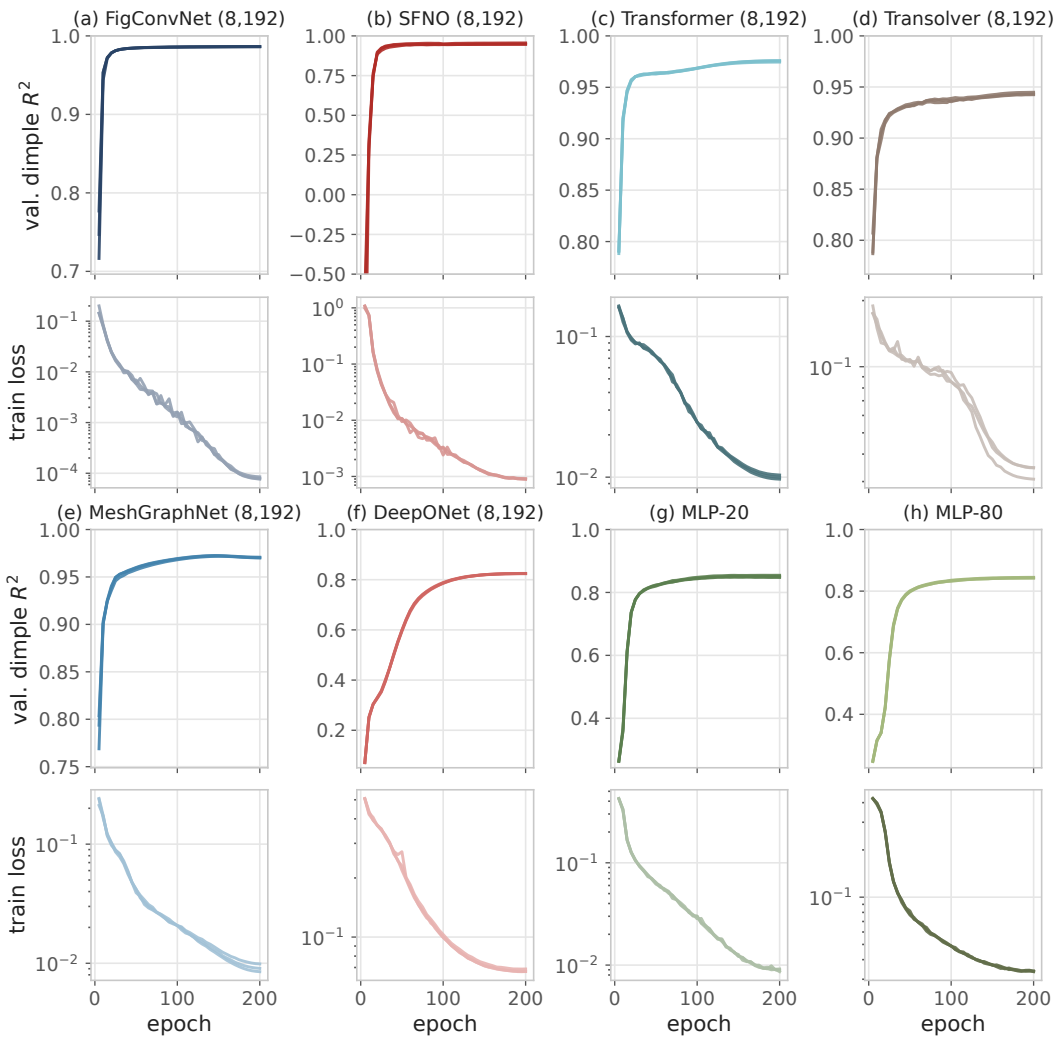


Figure 16: **Training and validation curves, B1**, for the final runs of every field model on 8,192 nodes.

and SFNO, with a median per-shell R^2 of -1.37 to -0.50. The failure is spread over the test set rather than carried by a few shells. DeepONet keeps a median of 0.29 on B3 with at most 1% of shells below zero, and it also has the lowest median on B1 (0.74).

What do the models predict for κ? On B1 the predicted κ follows the true one, with R^2_κ of 0.88 to 0.93 for FigConvNet, SFNO and the Transformer (Figure 21). On B3, FigConvNet and the Transformer predict almost the same κ for every dense shell, with prediction spreads of 0.005 to 0.009 against 0.031 for the truth and values close to the training mean of 0.458, while SFNO varies substantially across seeds. Across the three models, mean R^2_κ falls to -0.60 to -1.88, with SFNO ranging from -0.94 to -3.74 across seeds.

Where does the error sit? Figure 22 bins the test shells by the site ratio q. On B3 the 118 shells whose second site is weakest ($q < 0.5$) keep a positive median per-shell R^2 for FigConvNet, MeshGraphNet and Transolver in every seed (0.76, 0.82 and 0.27), and the median falls to -1.0 to -1.4 for q above 0.5. The κ error of the five models grows with q, from 3.2 to 4.6% below 0.5 to 9.1 to 12.4% above 0.9. The failure on B3 grows as the second site strengthens, toward the near-ties of Section 5.4.

Does the defect list do better than the surface? Figure 23 places every model that reads the list of defects beside the models that read w. On B1 the two inputs give comparable κ accuracy.

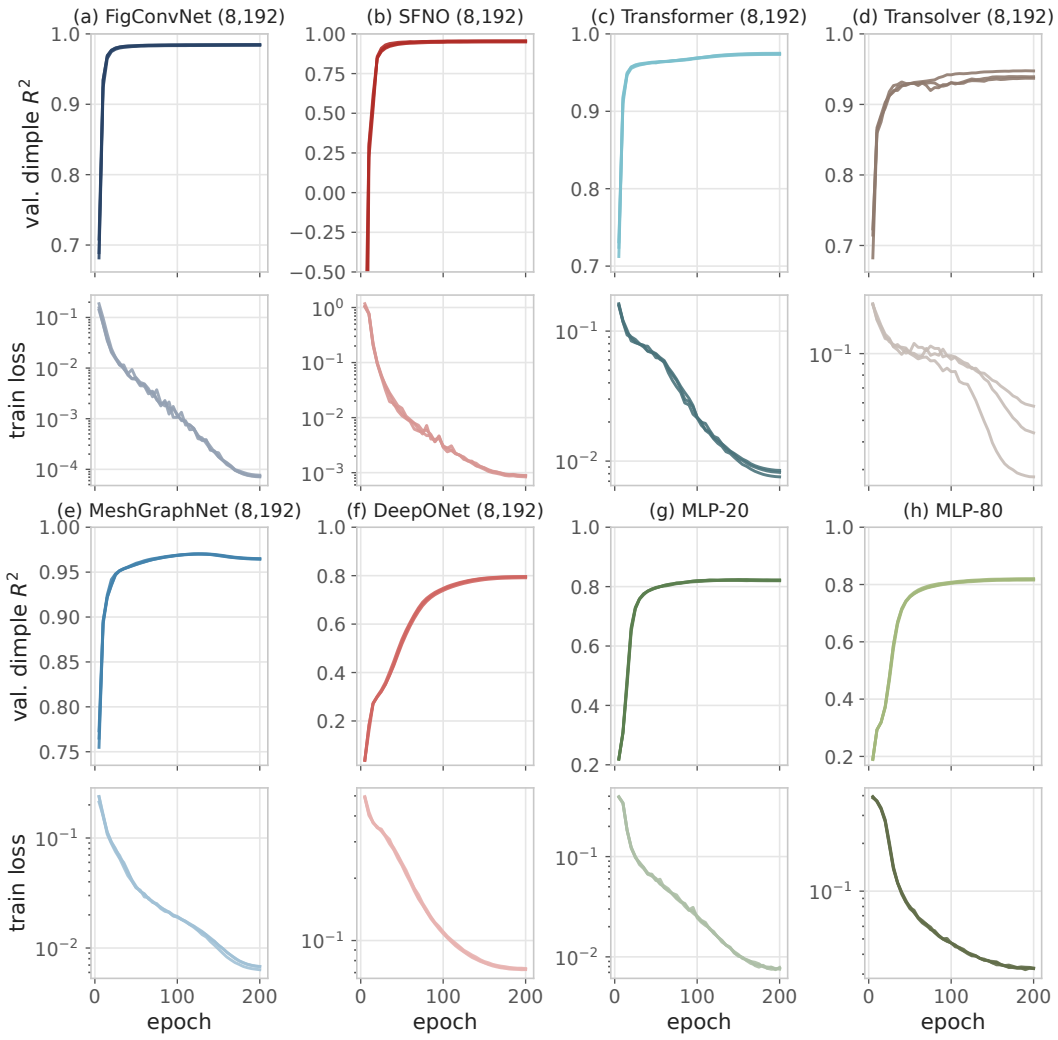


Figure 17: **Training and validation curves, B2**, for the final runs of every field model on 8,192 nodes.

Geometry models reach 0.58 to 1.32%, the frozen XGBoost 0.72%, the set models 0.81 to 0.93% and the fifteen tabular families 0.72 to 3.11%. On B3 every group moves to 4.7 to 9.8%, PCA-ridge on w to 31.8%, and only DeepONet (4.72%) and k-NN on w (4.86%) stay below the training mean (5.11%). The MLPs on the defect list diverge on B3 and are left out.

H.10 Intervention tests

The sweeps (App. E.5) change one factor of a core shell at a time and carry the solver's answer, so they test whether a trained model's prediction moves as the physics does, in the manner of behavioural testing (Ribeiro et al., 2020). Each sweep starts from a set of *source* shells and edits them one factor at a time. P1 rotates the whole defect pattern about the pole. An invariance test, the true κ does not change. P2 moves the deepest defect along a meridian. P3 brings two defects closer, from 25° to 10°. P4 deletes defects far from the two candidate sites. P2 to P4 are directional tests. Each edited version of a source is an *arm*. A model's *noise floor* is the change it predicts when a shell is only rotated. Models are applied without retraining. The detection and magnitude criteria were written before any sweep statistic was computed. At that time the B1 models' sweep predictions existed but had not been opened. One amendment replaced the detection rules (the noise floor built from analytic rotations of each test population, and one detection rule per sweep). Two clarifications (implementation details, and the P4 headline arm chosen from the solver's answer) followed. All three were recorded before any statistic was computed, and the clarifications change no criterion.

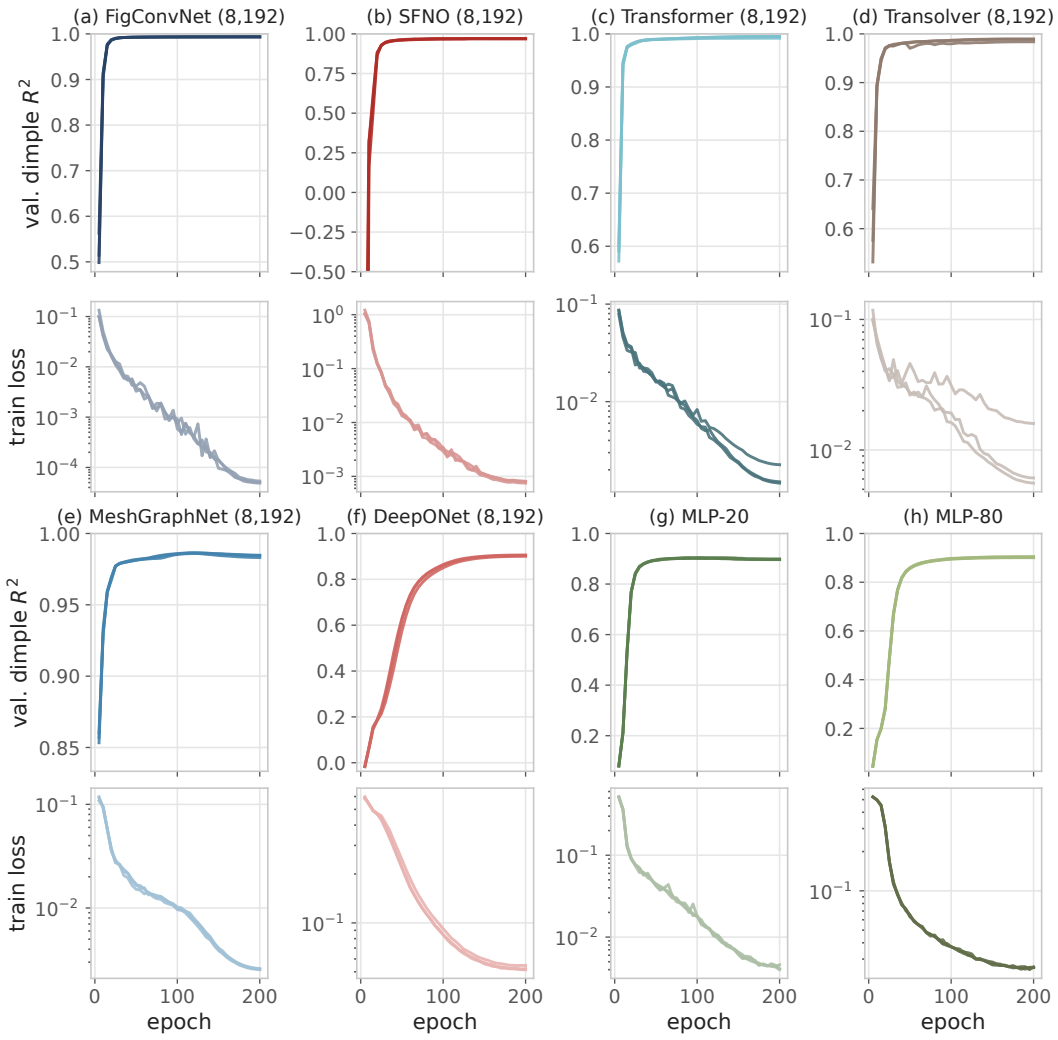


Figure 18: **Training and validation curves, B3**, for the final runs of every field model on 8,192 nodes.

Which models on which sweeps. The sources of P1 to P3 are 100 core shells, all in the B1 test set, so the B1 models see them for the first time. They are applied to P1, P2 and P3. The P4 sources are mostly B1 training shells (65 of 80), so B1 models are not used on P4. B3 models are applied to the 40 dense P4 sources, all in the B3 test set. The removal arms lie outside both the B3 training and test populations. B3 models are also applied to P3 as a second row, with a caveat. 84 of the 100 P3 sources are B3 training shells, so both their response and their noise floor are measured on shells at or near their training population.

Common rules. The unit is the source shell. Each change is paired per source against the sweep's reference arm (its unmoved arm for P2 and P4, with an identical-prediction check, and the $25°$ arm for P3, which differs from the source because the pair is moved to exactly $25°$), in absolute κ with percent printed. A model's noise floor J_m is the median absolute change of its seed-averaged prediction when the same source shells are rotated about the pole by 30, 60 and $90°$ (built from the defect table, no solve), on the population of the test (the P3 sources for P2 and P3, the 40 dense P4 sources for P4). Predictions are averaged over the three seeds before any statistic (the leaderboard instead averages metrics), per-seed medians must agree in sign, and intervals are percentiles of 2,000 source resamples. Cells are reported and left unranked. The smallest change a criterion could detect, J_m plus the half-width of the median-change interval, is printed as a heuristic.

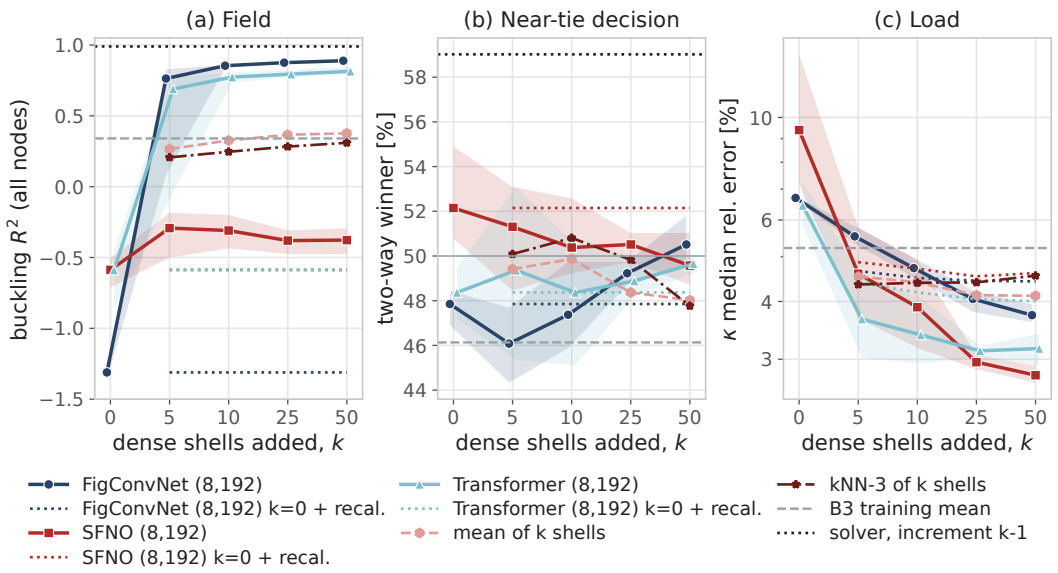


Figure 19: **Few-shot adaptation on B3.** Buckling R^2, near-tie winner rate and κ error against the number k of dense training shells, for the three fine-tuned models (8,192 nodes) and the three competitors without a network.

Table 24: **Few-shot adaptation on B3** (2,000 dense test shells, models on 8,192 nodes). Mean over cells (model seeds $\times$ draws) with the range. The recalibrated zero-shot row is the $k = 0$ model with a constant κ and uniform-shrink shift fitted on the k shells.

row	k	buckling R^2	near-tie	κ err [%]
FigConvNet	0	−1.311 (−1.362 to −1.270, 3)	0.48 (0.47 to 0.48, 3)	6.70 (6.65 to 6.75, 3)
FigConvNet	5	0.763 (0.079 to 0.830, 15)	0.46 (0.44 to 0.48, 15)	5.53 (5.34 to 5.82, 15)
FigConvNet	10	0.854 (0.839 to 0.858, 15)	0.47 (0.46 to 0.50, 15)	4.72 (4.51 to 4.97, 15)
FigConvNet	25	0.875 (0.874 to 0.878, 3)	0.49 (0.49 to 0.50, 3)	4.05 (3.80 to 4.22, 3)
FigConvNet	50	0.889 (0.888 to 0.889, 3)	0.51 (0.49 to 0.52, 3)	3.74 (3.61 to 3.94, 3)
FigConvNet zero-shot + recal.	5	−1.311 (−1.362 to −1.270, 15)	0.48 (0.47 to 0.48, 15)	4.65 (4.36 to 5.51, 15)
FigConvNet zero-shot + recal.	10	−1.311 (−1.362 to −1.270, 15)	0.48 (0.47 to 0.48, 15)	4.50 (4.35 to 5.04, 15)
FigConvNet zero-shot + recal.	25	−1.311 (−1.362 to −1.270, 3)	0.48 (0.47 to 0.48, 3)	4.43 (4.43 to 4.44, 3)
FigConvNet zero-shot + recal.	50	−1.311 (−1.362 to −1.270, 3)	0.48 (0.47 to 0.48, 3)	4.42 (4.39 to 4.46, 3)
SFNO	0	−0.587 (−0.710 to −0.503, 3)	0.52 (0.51 to 0.55, 3)	9.39 (7.16 to 13.69, 3)
SFNO	5	−0.292 (−0.501 to −0.184, 15)	0.51 (0.48 to 0.53, 15)	4.59 (3.67 to 5.79, 15)
SFNO	10	−0.309 (−0.435 to −0.200, 15)	0.50 (0.49 to 0.53, 15)	3.89 (3.17 to 4.92, 15)
SFNO	25	−0.381 (−0.475 to −0.308, 3)	0.51 (0.50 to 0.51, 3)	2.96 (2.85 to 3.07, 3)
SFNO	50	−0.377 (−0.477 to −0.294, 3)	0.50 (0.49 to 0.51, 3)	2.77 (2.67 to 2.90, 3)
SFNO zero-shot + recal.	5	−0.587 (−0.710 to −0.503, 15)	0.52 (0.51 to 0.55, 15)	4.86 (4.34 to 5.73, 15)
SFNO zero-shot + recal.	10	−0.587 (−0.710 to −0.503, 15)	0.52 (0.51 to 0.55, 15)	4.71 (4.26 to 5.99, 15)
SFNO zero-shot + recal.	25	−0.587 (−0.710 to −0.503, 3)	0.52 (0.51 to 0.55, 3)	4.53 (4.53 to 4.54, 3)
SFNO zero-shot + recal.	50	−0.587 (−0.710 to −0.503, 3)	0.52 (0.51 to 0.55, 3)	4.61 (4.50 to 4.75, 3)
Transformer	0	−0.589 (−0.857 to −0.454, 3)	0.48 (0.47 to 0.49, 3)	6.45 (5.84 to 7.23, 3)
Transformer	5	0.690 (−0.037 to 0.770, 15)	0.49 (0.45 to 0.53, 15)	3.66 (3.02 to 4.54, 15)
Transformer	10	0.774 (0.735 to 0.806, 15)	0.48 (0.45 to 0.51, 15)	3.39 (2.94 to 4.08, 15)
Transformer	25	0.795 (0.776 to 0.825, 3)	0.49 (0.48 to 0.50, 3)	3.13 (3.03 to 3.21, 3)
Transformer	50	0.815 (0.794 to 0.845, 3)	0.50 (0.49 to 0.50, 3)	3.17 (2.91 to 3.40, 3)
Transformer zero-shot + recal.	5	−0.589 (−0.857 to −0.454, 15)	0.48 (0.47 to 0.49, 15)	4.40 (4.06 to 5.00, 15)
Transformer zero-shot + recal.	10	−0.589 (−0.857 to −0.454, 15)	0.48 (0.47 to 0.49, 15)	4.18 (3.89 to 4.97, 15)
Transformer zero-shot + recal.	25	−0.589 (−0.857 to −0.454, 3)	0.48 (0.47 to 0.49, 3)	4.05 (3.99 to 4.12, 3)
Transformer zero-shot + recal.	50	−0.589 (−0.857 to −0.454, 3)	0.48 (0.47 to 0.49, 3)	4.01 (3.90 to 4.08, 3)
mean of the k shells	5	0.266 (0.159 to 0.295, 15)	0.49 (0.41 to 0.54, 15)	4.53 (3.89 to 5.94, 15)
mean of the k shells	10	0.326 (0.289 to 0.347, 15)	0.50 (0.46 to 0.54, 15)	4.40 (3.89 to 5.26, 15)
mean of the k shells	25	0.367 (0.365 to 0.369, 3)	0.48 (0.46 to 0.51, 3)	4.13 (3.93 to 4.25, 3)
mean of the k shells	50	0.377 (0.376 to 0.378, 3)	0.48 (0.48 to 0.48, 3)	4.11 (3.97 to 4.34, 3)
kNN-3 among the k shells	5	0.207 (0.145 to 0.244, 15)	0.50 (0.46 to 0.53, 15)	4.35 (3.68 to 5.35, 15)
kNN-3 among the k shells	10	0.246 (0.170 to 0.264, 15)	0.51 (0.45 to 0.55, 15)	4.39 (3.98 to 5.13, 15)
kNN-3 among the k shells	25	0.283 (0.276 to 0.293, 3)	0.50 (0.49 to 0.51, 3)	4.40 (4.21 to 4.75, 3)
kNN-3 among the k shells	50	0.310 (0.300 to 0.318, 3)	0.48 (0.47 to 0.49, 3)	4.55 (4.17 to 4.89, 3)
B3 training mean	–	0.340	0.46	5.22
solver, one step before the peak	–	0.990	0.59	–

P1, rigid rotation (invariance). P1 has no detection criterion. We report (i) $|\Delta\hat{\kappa}|$ per record against the solver's $|\Delta\kappa|$ (a median of 0.05 to 0.09%). (ii) field equivariance, the buckling R^2 of the prediction on the rotated shell, rotated back, against the prediction on the source, with a pure-interpolation calibration beside it (descriptive). (iii) the rate at which the predicted site follows the rotation (it lies within 4° of the rotated source site), split by whether the solver's own site followed.

Table 25: **Native track (76,805 nodes), B1**, predictions cut to the 8,192 scoring nodes and scored like every leaderboard row. Mean over three seeds (range).

model	κ err [%]	buckling R^2	RMSE [mm]	rel. L^2 [%]	clear $\leq 10^\circ$	near-tie
FigConvNet	0.91 (0.89–0.96)	0.987 (0.986–0.987)	0.0045 (0.0045–0.0045)	12 (12–12)	0.97 (0.97–0.97)	0.64 (0.63–0.66)
SFNO	0.58 (0.54–0.65)	0.890 (0.886–0.893)	0.0129 (0.0127–0.0131)	30 (27–32)	0.99 (0.99–0.99)	0.66 (0.64–0.67)
Transolver	0.67 (0.59–0.73)	0.928 (0.927–0.929)	0.0104 (0.0103–0.0104)	25 (25–26)	0.76 (0.74–0.77)	0.57 (0.54–0.60)
MeshGraphNet	0.82 (0.76–0.90)	0.923 (0.921–0.926)	0.0107 (0.0105–0.0109)	29 (29–30)	0.69 (0.69–0.70)	0.60 (0.59–0.61)

Table 26: **Native track (76,805 nodes) on B2 and B3**, predictions cut to the 8,192 scoring nodes. Mean over seeds (range).

model	protocol	seeds	κ err [%]	buckling R^2	RMSE [mm]	rel. L^2 [%]	clear $\leq 10^\circ$	near-tie
FigConvNet	B2	3	1.30 (1.24–1.37)	0.981 (0.981–0.982)	0.0069 (0.0068–0.0069)	14 (14–14)	0.83 (0.80–0.85)	0.54 (0.54–0.55)
FigConvNet	B3	3	6.60 (6.57–6.64)	−1.150 (−1.176 to −1.117)	0.0515 (0.0511–0.0518)	123 (122–124)	0.32 (0.32–0.32)	0.47 (0.47–0.48)
SFNO	B2	3	1.83 (1.78–1.90)	0.847 (0.838–0.855)	0.0196 (0.0191–0.0202)	35 (34–35)	0.80 (0.77–0.83)	0.56 (0.54–0.56)
SFNO	B3	3	7.89 (7.28–8.70)	−0.539 (−0.838 to −0.221)	0.0434 (0.0388–0.0476)	119 (109–127)	0.28 (0.27–0.29)	0.52 (0.51–0.53)
Transolver	B2	3	1.36 (1.30–1.42)	0.925 (0.924–0.926)	0.0137 (0.0137–0.0138)	29 (29–29)	0.32 (0.29–0.34)	0.55 (0.54–0.57)
Transolver	B3	3	6.46 (5.87–6.88)	−0.202 (−0.532 to 0.106)	0.0383 (0.0332–0.0435)	104 (92–117)	0.38 (0.36–0.42)	0.49 (0.46–0.50)

Table 27: **Node-resolution ladder, B1.** The same configuration trained on 2,048 or 8,192 nodes. The 8,192-node model is also scored on the 2,048 nodes of the smaller run, so the first two rows of a model compare like with like. Mean over three seeds (range).

model	trained / scored on	κ err [%]	buckling R^2	clear $\leq 10^\circ$	near-tie
FigConvNet	2,048 / 2,048	1.22 (1.16–1.30)	0.975 (0.973–0.976)	0.90 (0.88–0.91)	0.65 (0.64–0.66)
FigConvNet	8,192 / same 2,048	1.04 (0.96–1.10)	0.985 (0.985–0.985)	0.93 (0.92–0.93)	0.67 (0.66–0.68)
FigConvNet	8,192 / 8,192	1.04 (0.96–1.10)	0.985 (0.985–0.985)	0.96 (0.96–0.96)	0.60 (0.60–0.62)
MeshGraphNet	2,048 / 2,048	0.85 (0.82–0.88)	0.961 (0.960–0.961)	0.81 (0.79–0.84)	0.64 (0.62–0.68)
MeshGraphNet	8,192 / same 2,048	0.70 (0.69–0.72)	0.968 (0.968–0.969)	0.86 (0.86–0.88)	0.60 (0.58–0.62)
MeshGraphNet	8,192 / 8,192	0.70 (0.69–0.72)	0.968 (0.968–0.968)	0.89 (0.88–0.92)	0.57 (0.52–0.61)
Transformer	2,048 / 2,048	1.20 (1.13–1.28)	0.934 (0.933–0.935)	0.65 (0.64–0.66)	0.57 (0.52–0.60)
Transformer	8,192 / same 2,048	0.65 (0.62–0.70)	0.970 (0.969–0.970)	0.87 (0.85–0.88)	0.68 (0.66–0.70)
Transformer	8,192 / 8,192	0.65 (0.62–0.70)	0.969 (0.969–0.970)	0.88 (0.85–0.90)	0.65 (0.61–0.68)
Transolver	2,048 / 2,048	1.51 (1.39–1.66)	0.898 (0.897–0.900)	0.54 (0.52–0.55)	0.56 (0.55–0.57)
Transolver	8,192 / same 2,048	0.91 (0.79–0.99)	0.923 (0.921–0.927)	0.70 (0.70–0.70)	0.60 (0.58–0.64)
Transolver	8,192 / 8,192	0.91 (0.79–0.99)	0.922 (0.920–0.926)	0.72 (0.71–0.74)	0.55 (0.53–0.56)

Table 28: **SFNO input resolution.** The reported 180×720 SFNO against the same operator on a 64×128 grid, which has as many samples as the 8,192 scoring nodes (periodic longitude). The pure control retrains the reported configuration on B1. The equivalent search re-runs the B1 search at 64×128 (25 trials, the top two retrained for 200 epochs) and trains the winner on B2 and B3. Mean over three seeds with the range.

	row	seeds	κ err [%]	buckling R^2	clear $\leq 10^\circ$	near-tie
B1	180×720 (reported)	3	0.58 (0.57–0.60)	0.934 (0.929–0.941)	0.98 (0.97–0.98)	0.65 (0.64–0.66)
B1	64×128 pure control	3	0.62 (0.61–0.62)	0.880 (0.876–0.888)	0.96 (0.96–0.97)	0.61 (0.59–0.63)
B1	64×128 equivalent search	3	0.53 (0.50–0.55)	0.895 (0.888–0.899)	0.98 (0.97–0.99)	0.65 (0.64–0.66)
B2	180×720 (reported)	3	1.29 (1.15–1.43)	0.915 (0.904–0.923)	0.75 (0.73–0.76)	0.56 (0.55–0.57)
B2	64×128 equivalent search	3	1.75 (1.63–1.95)	0.812 (0.808–0.814)	0.82 (0.82–0.83)	0.56 (0.55–0.57)
B3	180×720 (reported)	3	9.37 (7.16–13.64)	−0.587 (−0.709 to −0.505)	0.30 (0.28–0.30)	0.52 (0.50–0.54)
B3	64×128 equivalent search	3	6.16 (6.06–6.32)	−0.713 (−0.801 to −0.588)	0.30 (0.28–0.30)	0.51 (0.50–0.52)
B3	training mean	–	5.11	0.340	0.00	0.47

Built-in azimuthal equivariance is marked per model. SFNO on the 180×720 grid has it up to interpolation. The node models do not. FigConvNet does not (it convolves on three axis-aligned factorised grids with separate weights and absolute position encodings, so no rotation about the pole other than the identity is a symmetry of the architecture). The 64×128 SFNO control is not used here.

P2, moving the deepest defect (direction). On the 87 analysed sources, the pooled least-squares slope, with intercept, of $\Delta\hat{\kappa}$ on $\Delta\kappa$ over the moved arms (reference-arm points excluded). A response is detected if the source-bootstrap 95% interval of the slope excludes 0 and its sign agrees in all three seeds. The slope's distance from 1 is printed. The per-source Spearman correlation is descriptive. P2 has no magnitude criterion, and its site is left unscored (fixed with the criteria).

P3, one pair from 25° to 10° (direction). At every separation, the paired change against the 25° arm. A response is detected if the source-bootstrap 95% interval of the median signed paired change

Table 29: **Tabular families on B1** (list of defects, 81 deepest defects, one Optuna search per family on B1 validation with the median κ error, three refits). Test median relative error of κ and R^2_κ, mean $\pm$ s.d. over seeds.

family	trials	κ err [%]	R^2_κ
XGBoost	175	0.72 ± 0.01	0.888
hist. gradient boosting	125	0.73 ± 0.00	0.886
LightGBM	175	0.78 ± 0.02	0.890
extra trees	100	0.83 ± 0.01	0.852
random forest	100	0.83 ± 0.00	0.852
FT-Transformer	41	0.84 ± 0.02	0.872
CatBoost	40	0.95 ± 0.05	0.885
ResNet	40	0.98 ± 0.08	0.852
MLP	40	1.07 ± 0.04	0.818
SVR	75	1.55 ± 0.00	0.612
Gaussian process	50	1.84 ± 0.00	0.696
TabNet	40	1.93 ± 0.38	0.749
k-NN	75	1.96 ± 0.00	0.589
elastic net	50	3.04 ± 0.00	0.550
ridge	40	3.11 ± 0.00	0.550

Table 30: **Tabular models on the three protocols** (median relative error of κ [%], R^2_κ in brackets, three seeds). The frozen XGBoost is the B1 configuration refitted on each split. All-family searches and set models are tuned on each split's own validation set. Set models predict $\kappa - \kappa_{\mathrm{WL}}$ unless marked.

model	B1	B2	B3
XGBoost, B1 configuration frozen	0.72 (0.888)	1.69 (0.042)	6.50 (−0.703)
all families, list of defects	CatBoost: 0.79 (0.890)	XGBoost: 1.68 (−0.002)	XGBoost: 6.44 (−0.647)
all families, physical features	extra trees: 0.86 (0.835)	extra trees: 1.60 (−0.061)	extra trees: 6.35 (−0.609)
all families, interaction features	random forest: 0.70 (0.849)	random forest: 1.52 (−0.033)	XGBoost: 7.72 (−1.199)
flat set model, core only	0.87 (0.866)	1.37 (0.377)	8.28 (−1.226)
flat set model, with singles and doubles	0.88 (0.872)	1.63 (0.157)	6.19 (−1.841)
DeepSets set model, core only	0.81 (0.848)	1.68 (−0.353)	9.83 (−2.097)
DeepSets set model, with singles and doubles	0.84 (0.849)	1.80 (0.242)	10.99 (−2.775)
pairwise set model, core only	0.93 (0.849)	2.80 (−0.537)	9.30 (−2.370)
pairwise set model, with singles and doubles	0.94 (0.843)	1.79 (0.246)	8.85 (−1.577)
flat set model, target κ (with extra)	1.07 (0.835)	–	–
DeepSets set model, target κ (with extra)	0.92 (0.850)	–	–
pairwise set model, target κ (with extra)	1.06 (0.846)	–	–
flat set model, trained on singles and doubles only, tested on the core (4,399 shells)	21.24 (−28.969)		
DeepSets set model, trained on singles and doubles only, tested on the core (4,399 shells)	46.37 (−70.661)		
pairwise set model, trained on singles and doubles only, tested on the core (4,399 shells)	37.56 (−32.153)		

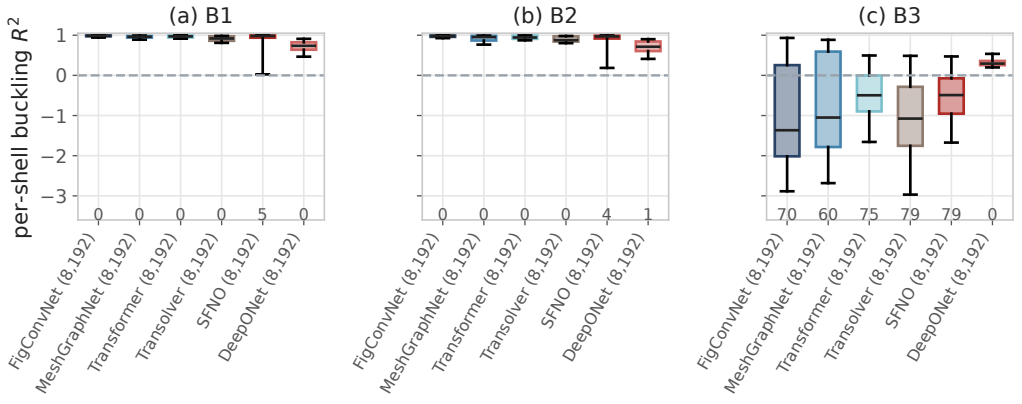


Figure 20: **Per-shell buckling** R^2, models on 8,192 nodes, clipped at -3. Each shell contributes one value per training seed, and boxes pool the three seeds (quartiles, whiskers the 5th to 95th percentiles). Numbers give the share of shells below zero [%], the mean over seeds.

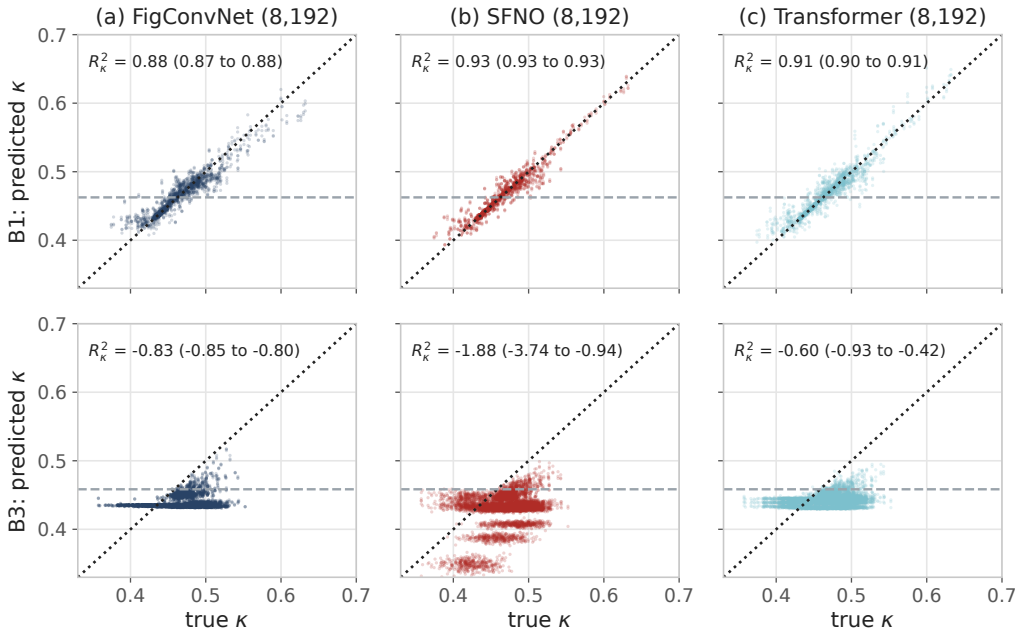


Figure 21: **Predicted against true** κ on B1 and B3, models on 8,192 nodes, the three training seeds overlaid. Each panel gives R^2_κ as the mean over seeds with the range. Dotted line, perfect prediction. Dashed line, the training mean of the protocol.

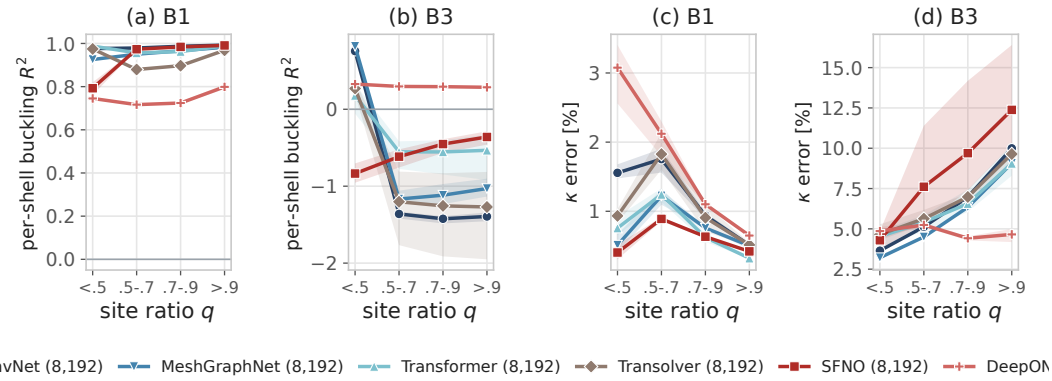


Figure 22: **Error against the site ratio** q**.** Median per-shell buckling R^2 (a, b) and median κ error (c, d) in four bins of q, models on 8,192 nodes. Lines give the mean over the three training seeds of the per-seed medians, and bands give the range over seeds. B3 holds 118, 656, 998 and 428 shells in the four bins.

lies entirely outside $[-J_m, +J_m]$ and its sign agrees in all three seeds. The magnitude is recovered if the median over sources of the ratio $\Delta\hat{\kappa}/\Delta\kappa$ lies in $[0.5, 2]$, over sources whose true change exceeds the solver floor (0.09% of the reference κ). A negative ratio is a wrong sign and is not recovered. The headline arm is 10°, with the share of sources recovering at least half the true drop. For the site, we report the share of sources whose predicted site at 10° lies more than 2° from the centre of every listed defect, against the solver's 97% under the same rule.

P4, removing far defects (direction, B3 models). The same detection and magnitude rules as P3, against the all-kept arm, on the 40 dense sources. The headline arm keeps 50% of the far defects, chosen from the solver's answer (its largest median drop, -4.8%) before any model statistic. All four removal arms are reported. We also report the sign agreement rate, and the site, which moves if the predicted site is more than 4° from the prediction on the all-kept arm (the solver's own threshold),



Figure 23: κ **error by input.** Median relative error of every model on B1, B2 and B3 (Tables 2, 18 to 20, 29 and 30). Geometry models on 8,192 nodes. The fifteen tabular families were run on B1 only. The diverged MLPs are left out on B3.

computed on the solver's 98 dropping arms (true drop $> 1\%$) and set against the solver's own rate under the same 4° rule, 95 of the 98 (model and solver on the same arms). The recorded reference, 92 of 98, counts only moves outside both protected zones. The rate on all 160 removal arms is printed beside it, with the solver's rate on the same arms.

Results. All numbers below come from `benchmark/analyses/interventions_score.py` (released with the scorer), with the criteria above. *P1.* The solver changes κ by a median of 0.05 to 0.09%, and its failure site follows the rotation in 214 of 300 rotations under the 4° rule. The predicted site follows the rotation on 72% of rotations for SFNO, 17 to 51% for the other geometry models and 2% for the two MLPs on the list of defects. *P2.* A response is detected for MeshGraphNet, SFNO and the Transformer, with pooled slopes of 0.04 to 0.14 against 1 for a model that follows the solver. The other B1 models show no detected response. *P3.* At 10° the solver's median change of κ is -14.2% over the 100 sources. The B1-trained models predict median changes between -1.05 and $+0.32\%$, and at most 3% of the sources recover half of the true drop. Where a response is detected (MeshGraphNet, SFNO and the Transformer on B1, SFNO on B3), it is far too small, and for SFNO and the Transformer it has the wrong sign. The predicted site lies more than 2° from every listed defect on 7 to 33% of the sources for the B1 models other than SFNO (62%), against 97% for the solver. *P4.* On the dense sources the solver lowers κ when half of the far defects are removed. Every B3 model misses the magnitude of this change. DeepONet responds with the sign of the solver, a median change of -2.2% against -4.8%, which is less than half of the true drop. A response is also detected for SFNO with the wrong sign and for the diverged MLP-80, and the other models show no detected response. The full per-arm statistics are released in `score_sweeps.json`.

H.11 Precision of the labels

Near an instability the simulated label is itself imprecise. Table 31 scores the released solver states one step before and one step after the peak as if they were models. A model that matches the peak as closely as they do is within one step of the solver. Rotating the defect pattern leaves the physics unchanged, and the solver keeps its failure site on 72% of 300 rotations of B1 near-ties. On a four times finer mesh, the failure site moved on 2 of 40 B1 near-ties and on 0 of 40 clear-winner shells, with a one-sided Fisher $p = 0.25$, and κ changed by a median of 0.26% and 0.42% (App. E.6.1).

I Why does the near-tie decision stay close to chance from sparse to dense?

Near-tie decision within distribution. Trained on B1, where about half of the training shells are dense (49.5%), SFNO, the Transformer and FigConvNet choose the true winner on 59 to 63% of

Table 31: **Precision of the labels.** The released solver states one step before and one step after the peak, scored against the peak state like a model. Rotating the whole defect pattern (P1) keeps the same failure site in 72% of 300 rotations of 100 B1 near-ties.

	B1		B2		B3	
metric	before	after	before	after	before	after
buckling R^2	0.993	0.966	0.994	0.966	0.990	0.931
site within 10° [%]	100	100	98	100	99	100
near-tie winner rate [%]	69	80	78	82	60	70

dense near-ties. The solver itself keeps its failure site on 72% of rotated near-ties. In distribution the decision is therefore learned above chance, and close to the precision of the label.

Near-tie labels on B3. The solver state one step before the peak agrees with the peak state on 60% of B3 near-ties, and the state one step after on 70% (Table 31). The dense near-ties of B3 are therefore less sharply defined than those of B1 (69% and 80%). A model at 50% is still far below this level.

Field and near-tie decision after fine-tuning. After fine-tuning on 50 dense shells, FigConvNet reaches a buckling R^2 of 0.89 and the Transformer 0.82, and their near-tie winner rates stay at 51% and 50% (Table 24). The competitors without a network score 48 to 51%. The deeper of the two candidate sites wins on 48% of B3 near-ties. Depth alone therefore does not decide the winner in dense shells, and a few dense examples are not enough to learn what does.

Clear winners on B3. On the B3 clear-winner shells every geometry model places the failure site within 10° on only 30 to 32% of shells (DeepONet on 25%), below the 43% of the deepest point of w (Table 2). The models carry a rule learned on sparse shells, where the deepest defect is the failure site on all 94 sparse clear-winner B1 shells, into a regime where it holds on 38% of the dense ones.

J Does the load need the full geometry?

Knockdown factor from the list of defects. On B1 the fifteen tabular families reach a median κ error between 0.72% (XGBoost) and 3.11% (ridge regression), and a random forest on interaction features reaches 0.70% (Tables 29 and 30). The set models reach 0.81 to 0.93%. The geometry models reach 0.58 to 1.04%, except DeepONet at 1.32%, and the MLPs on the list of defects 1.19 and 1.20%. The best tree models on the list of defects are therefore close to the best geometry models (SFNO 0.58%, Transformer 0.65%, MeshGraphNet 0.70%), and gradient boosting on simple features of w reaches 1.12%.

Weakest-link rule on sparse shells. The weakest-link rule reaches 1.14% on sparse B1 shells and 5.89% on dense ones (App. F). The load of a sparse shell is set mainly by its worst defect. The load of a dense shell depends on the interaction between defects, which the list of defects encodes only through the positions.

Across the shifts. On B2 the frozen XGBoost reaches 1.69%, the other tabular models 1.37 to 2.80%, the 80-defect MLP 2.36% and the geometry models 1.08 to 1.62%, except DeepONet at 2.38%. On B3 every tabular model reaches 6.19 to 10.99%, both MLPs diverge, and the geometry models reach 4.7 to 9.4% against 5.11% for the training mean, which only DeepONet beats. The load is therefore easy in distribution and hard from sparse to dense, for every input.

Composing single- and two-defect shells. Set models trained on the single- and two-defect shells alone and tested on the 4,399 core shells reach a median κ error of 21 to 46%. Adding these shells to the core training set changes the B1 error by at most 0.03 points (Table 30).

K How does this dataset compare with public benchmarks?

Structural-mechanics datasets. Datasets in structural mechanics involve smooth solid-mechanics fields (Pfaff et al., 2021; Lejeune, 2020; Li et al., 2023). ConcreteShellFEA (Pollet

Table 32: BLISS against the nearest simulation benchmarks for machine learning. ✓ the dataset's own paper or documentation states the property, (✓) it holds with a stated qualification, × its documentation reports the property as absent or the released artefacts demonstrably lack it, **?** we could not verify it either way. BLISS is the only row with a full mark on instability and on ambiguity together. ABC columns and the Voronoi fracture paths come nearest, each releasing the outcome of a selection without a margin between the competing branches. BLISS is not alone on experimental validation, and it does not lead on irregular geometry or on physical splits. Per-cell evidence and sources in Appendix K.

	exp. validation	nonlinear instability	irregular geometry	field target	scalar target	physical OOD split	ambiguity labels	reprod. generator	data licence
PDE and fluid benchmark suites									
PDEBench	×	×	×	✓	×	×	×	✓	CC BY 4.0
The Well	×	(✓)[a]	×	✓	×	×	×	(✓)[b]	CC BY 4.0
FD-Bench	×	×	(✓)[c]	✓	×	(✓)[d]	×	**?**	unknown[e]
shape-varying engineering surrogates									
AirfRANS	✓[f]	×	✓	✓	✓	✓[g]	×	✓	ODbL 1.0
DrivAerML	✓[h]	×	✓	✓	✓	✓[i]	×	(✓)[j]	CC BY-SA 4.0
DrivAerNet++	✓[k]	×	✓	✓	✓	×[l]	×	(✓)[m]	CC BY-NC 4.0
SIMSHIFT	×	×	✓	✓	×	✓[n]	×	×[o]	MIT
solid mechanics and structural instability									
MGN deforming plate	×	×	✓	✓	×	×	×	×[p]	unstated[q]
Mechanical MNIST + Distribution Shift	×	×	×	✓	✓	(✓)[r]	×	✓	CC BY-SA
Geo-FNO elas./plas.	×	×	(✓)[s]	✓	×	×	×	×[t]	unknown[u]
ABC columns	×	✓	✓	×	×[v]	×	(✓)[w]	✓	unknown[x]
Voronoi fracture paths	×[y]	(✓)[z]	✓	(✓)[aa]	×	×	(✓)[ab]	✓	unknown[ac]
ConcreteShellFEA	×	(✓)[ad]	(✓)[ae]	✓	✓	×	×	×	CC BY 4.0
BLISS (ours)	✓[af]	✓	(✓)[ag]	✓	✓	✓	✓	(✓)[ah]	CC BY 4.0 / MIT

Column definitions, per-cell notes and the evidence for every mark are in Appendix K.

et al., 2026a) adds shell buckling at scale, with a nonlinear subset whose fields at the critical point we could not verify from the release.

Competing outcomes and label precision in related work. Critical events can introduce targets that are more challenging than smooth field regression. Discontinuous targets can reduce attainable approximation rates (Petersen & Voigtlaender, 2018; Lanthaler et al., 2023), and multimodal or uncertainty-aware predictors are natural tools when outcomes compete (Bishop, 1994; Lakshminarayanan et al., 2017; Kohl et al., 2018; Lippe et al., 2023; Lipman et al., 2023; Hendriks et al., 2025). They are rarely scored on the selected outcome, because benchmarks rarely release which outcomes competed or by how much. Selective prediction studies when to withhold a prediction (Chow, 1970; El-Yaniv & Wiener, 2010; Geifman & El-Yaniv, 2019; Cortes et al., 2016). Uncertain labels are often handled by keeping the disagreement between annotators instead of reducing it to one label (Sheng et al., 2008; Peterson et al., 2019; Beyer et al., 2020; Plank, 2022). In simulation, the label carries discretisation and solver error (Roache, 1998; Oberkampf & Roy, 2010), and near an instability this error can decide the outcome. BLISS measures the precision of its labels with neighbouring solver states, a rotation test and a finer mesh, and reads near-tie decisions against it (Section 4.6).

Table 1 in the main text compares BLISS with public benchmarks on a few axes. Table 32 extends the comparison to nine properties, with the evidence for each cell.

Every mark follows the rules below, and the notes give the evidence for each qualified mark. A mark is given only where the dataset's own paper or documentation states the property. A cross

is given where that documentation reports the property as absent or where the released artefacts demonstrably lack it. Where neither holds, the cell is left unknown rather than assumed absent.

Three row names are shorthand. *Mechanical MNIST + Distribution Shift* names two releases together, the base dataset (Lejeune, 2020) and its companion (Yuan et al., 2022), because the shifted splits exist only in the second. *Geo-FNO elas./plas.* is the elasticity and plasticity pair released with the Geo-FNO operator (Li et al., 2023). *Voronoi fracture paths* is the disordered-lattice fracture dataset of Karapiperis & Kochmann (2023), the closest non-buckling benchmark whose target is decided by a selection. The equivariant flow-matching work cited in the paragraph above (Hendriks et al., 2025) has no row. It releases a method and its code, and no dataset.

Column definitions. *exp. validation*. At least one released target quantity of this simulation pipeline has been compared against physical measurements of the same quantity (the notes give the scope, which is partial in every case). *nonlinear instability*. The released target is decided at a loss of stability (limit point or bifurcation) of a geometrically nonlinear simulation. *irregular geometry*. The domain is non-Cartesian, discretised on an unstructured mesh or point cloud, and its shape varies between samples. *field target*. A spatially resolved field is a released prediction target. *scalar target*. A scalar quantity of interest is a released prediction target. *physical OOD split*. The release fixes a train/test split along a physical parameter rather than at random. *ambiguity labels*. A per-sample margin between competing outcomes is released, so accuracy can be stratified by it. *reprod. generator*. The release contains the inputs and scripts needed to regenerate a sample. *data licence*. The licence attached to the data. The code licence is left out.

Notes. *a* two datasets simulate hydrodynamic instabilities, but the target is the time-evolving field. These datasets have no critical load. *b* each dataset README links the external repository and solver that produced it, and the benchmark ships without a generator. *c* the ten headline scenarios are on regular grids. Non-Cartesian boundaries and mesh/particle discretisations appear in the supplementary tasks. *d* the evidence is zero-shot evaluation on out-of-distribution initial conditions, resolutions and rollout horizons, which is a generalisation protocol. The release fixes no split of its own samples along a physical parameter, so the mark is partial. *e* the code repository carries CC BY-NC-SA 4.0 while the Hugging Face repositories holding the self-generated trajectories declare MIT. The two disagree on commercial use and neither is stated as the licence of the benchmark data as a whole, so the cell is unknown. *f* pressure coefficients and lift/drag polars of the CFD setup against NASA wind-tunnel data for the NACA 0012 and 4412, two reference airfoils outside the 1,000 released cases. *g* Reynolds-number and angle-of-attack extrapolation splits, plus two data-scarcity regimes. *h* surface pressure and forces against published wind-tunnel data for the baseline geometry only, not for the 484 released morphed variants. *i* four out-of-distribution regimes (geometry, high-drag, low-drag, rear separation) added to the dataset's documentation site after the 2024 paper. *j* OpenFOAM meshes for every case and a documented automated workflow are supplied. We could not verify that a runnable case setup is public. *k* drag coefficient against wind-tunnel data for two baseline cars. *l* one train/validation/test list of design ids, and the release holds only in-distribution splits. *m* parametric geometry models are released. The CFD case setup is not in the repository. *n* four tasks, each split into non-overlapping regions of the parameter space at three difficulty levels. *o* four preprocessed archives plus training and split code, without a solver setup. *p* pre-generated TFRecords only. The COMSOL and ArcSim setups are not released. *q* model code is Apache 2.0, while the data carry no stated licence on the storage bucket, in the README, on the project page or in the paper, so the cell reads unstated rather than none, which would read as a decision. *r* the base release defines no shifted split. The row names both releases because the covariate-shift, mechanism-shift and sampling-bias splits are defined only in the companion release Mechanical MNIST: Distribution Shift (Yuan et al., 2022). *s* the elasticity set is an unstructured point cloud of about 1,000 points around a void whose radius field is resampled per sample. The plasticity set is a fixed 101×31 structured mesh in which only the shape of the impacting die varies. *t* the public repository ships model and training scripts only, without data-generation code, and the plasticity data were produced with a commercial finite element solver. *u* the code repository is MIT. The four datasets are distributed through a shared cloud folder with no licence attached to them. *v* the released label is the binary buckling direction. The dataset has no continuous scalar target. *w* releases the binary outcome of a symmetry-breaking selection, the nearest analogue, but no per-sample margin between the competing branches. The confidence studied there is a property of the trained ensemble and is absent from the dataset labels. *x* the paper states only that the dataset is released "under open source licenses" and names none. The hosting record returns HTTP 403 to automated fetches and no archived snapshot exists, and the repository's MIT licence covers the code. *y* the elastoplastic beam model is calibrated and validated against tensile tests of additively manufactured samples, but the released target, the crack path, is never compared with a measured crack path. *z* the crack path is a selection among competing beams of a geometrically nonlinear corotational elastoplastic model, but each advance is set by a strut failure criterion instead of a limit point or bifurcation of the equilibrium path. *aa* the target is spatially resolved on the lattice, as the sequence of broken edges, but no continuum field (displacement, stress) is released. *ab* the outcome of a path selection is released, and the paper reports that competing directions become equivalent as disorder vanishes, but no per-sample margin between the competing paths is released. The softmax confidence discussed there is an output of the trained model. *ac* the deposit's DataCite record carries an empty rights list and the repository landing page returns HTTP 500 to automated fetches. The model code is MIT. *ad* 5,000 of

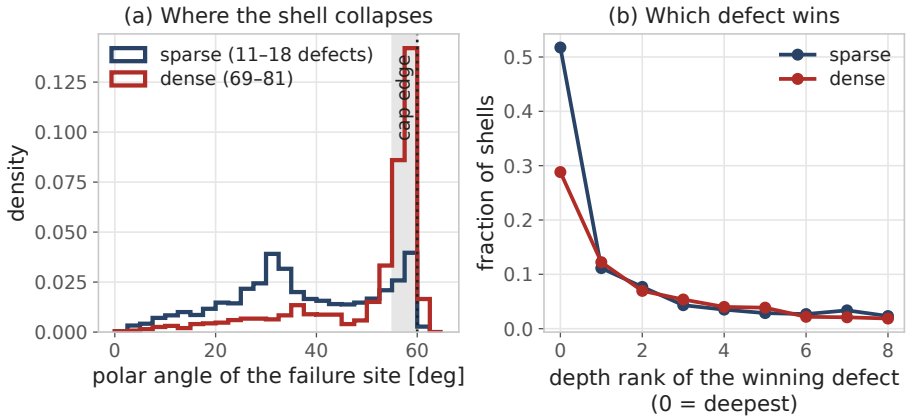


Figure 24: **The failure site against the cap of the defect generator. a** Polar angle of the failure site by regime, with the outer 5° of the 60° cap shaded. **b** Depth rank of the defect nearest the failure site.

the 45,000 shells are nonlinear buckling analyses. The other 40,000 are linear eigenvalue analyses. *ae* the shells are curved and their span, height and thickness vary per sample, but the record does not document the mesh, so the unstructured-discretisation half of the column could not be verified. The mark is partial for that reason, as ours is for a different one. *af* the collapse pressure of eight μCT-characterised specimens, simulated on their measured geometry (App. E.6.3). The field target is verified by mesh convergence. The specimens differ from the released shells. *ag* an unstructured shell mesh on a curved domain, but one mesh topology is shared by all shells, which differ only in their defects. A map onto a spherical grid is also released. *ah* the lists of defects and the deck generator are released and regenerate any shell in a median of about two core-hours (eight cores), but running them needs a commercial solver licence.

L LIMITATIONS

BLISS is narrow by design, so that changes in model performance can be attributed more directly to the geometry of the defects. The defects are smooth inward defects of one width, on a shell of uniform thickness, with one radius-to-thickness ratio, one material, one boundary condition and one load. Real shells also carry bumps and a thickness field, which a scan provides and the input of this benchmark leaves out. The displacement field is verified by mesh refinement and is not measured, while the collapse load is checked against eight physical tests. At a near-tie the failure site is at the edge of the precision of the solver, so we score those shells against that precision instead of removing them. The number of defects and their spacing change together between the two regimes, so B3 is a transfer between an isolated and a crowded defect pattern. The intervention test P3 changes the spacing of one pair alone, and it shows the same abrupt change, but B3 itself mixes the two factors. The solver is proprietary, although the release carries everything needed to rebuild its input. Our Transformer is a plain reference and is not a published surrogate. The few-shot shells are drawn from the same population as the test shells, and one fine-tuning recipe is used.

Failure-site position in dense shells. Defect centres are drawn within 60° of the pole. Dense shells collapse near the edge of this cap far more often than sparse shells (Figure 24), and on a defect that is often not the deepest. This is a shift in where collapse starts within the region where defects are placed, and it is part of what B3 measures. It is a shift of the sampled geometry and involves no boundary physics. The cap of 60° follows Derveni et al. (2023), whose simulations of one defect at different positions found the knockdown factor unchanged for any position within 60° of the pole, with the clamped rim mattering only closer to the equator. A release with defects spread more widely would separate the position of collapse from defect interaction, and it is a natural next version.

M OPEN PROBLEMS

Learning the near-tie decision from few examples. In distribution the models choose the failure site of a near-tie at 59 to 65%, below the 72% at which the solver agrees with itself. From sparse to dense they stay close to chance, even after 50 dense examples recover the field of FigConvNet and the Transformer (App. I). A model that learns the decision from few examples, or that transfers it from sparse to dense shells, would close the main gap that BLISS exposes.

Using the released site ratio. The site ratio q ships with every shell. A model can be trained to abstain above a threshold of q, to rank the two candidate sites, or to be calibrated where the label is less sharp. Mixture, ranking and generative models are natural candidates.

Finding what decides a near-tie. The deeper of the two candidate sites wins on about half of the near-ties of every protocol (App. F.1). Depth ordering therefore does not decide the winner. The interaction between defects, the position relative to the clamped rim and the local curvature are the next candidates, and the released site ratio makes them testable.

Carrying the thickness. A scanned shell provides a thickness field, which the input of this benchmark leaves out.

N DATASHEET FOR DATASETS

This appendix follows the Gebru et al. (2021) template. Headings are preserved verbatim where possible, and the full questions and answers are reproduced below. The same datasheet ships with the release.

N.1 MOTIVATION

For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description. BLISS was created to study learning of a collapse from geometry. The task is to predict, from the surface of an imperfect hemispherical shell, its knockdown factor κ (how much the defects lower the collapse load), its displacements at collapse and the site where the collapse starts (Section 4.1). The gap is that existing scientific-ML benchmarks create their distribution shifts from random splits, new initial conditions or held-out ranges of sampled inputs, while in BLISS a change in the physical mechanism creates the shift. To our knowledge, BLISS is the first dataset that pairs the three targets for one nonlinear instability (Section 2).

Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)? The authors of this paper. Their identities and affiliations are withheld for anonymous review. All simulations, including the single- and two-defect reference shells, were performed by the authors.

Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number. Funding is disclosed in the final version (withheld for anonymous review).

Any other comments? None.

N.2 COMPOSITION

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? Are there multiple types of instances (e.g., movies, users, and ratings; people and interactions between them; nodes and edges)? Please provide a description. Each instance is one simulated imperfect hemispherical shell under external pressure, solved up to and past the peak of its load. There are three types (Table 9). Core shells carry 11 to 81 defects, with their surface, fields and labels. Intervention (sweep) shells are core shells edited one factor at a time, with the same content and the id of their source. Single- and two-defect reference shells are parametric rows, the defect parameters and κ, without fields.

How many instances are there in total (of each type, if appropriate)? 8,528 shells, namely 4,399 core shells (2,199 sparse and 2,200 dense), 1,870 sweep shells (P1 to P4) and 2,259 single- and two-defect reference shells. The 6,269 shells with fields take about 16 GB for the core.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? If the dataset is a sample, then what is the larger set? Is the sample representative of the larger set (e.g., geographic coverage)? It is a sample. The larger set is every pattern of inward Gaussian defects on one shell geometry. The core samples two design populations of multi-defect shells, with a minimum centre separation of 25° (sparse) and 10° (dense), and defect centres within 60° of the pole (Appendix E.2). It is representative of these two populations only. It does not cover other slendernesses, defect widths, boundary conditions or loadings.

What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description. Raw solver output on one base mesh of 76,805 nodes, common to all shells. The input is the surface deviation w at every node. The targets are κ and the displacements at the peak load. The record also holds the states one simulation step before and after the peak, the buckling mode, the load history and three per-shell numbers (Appendix E.3, Table 5). Radius, wall thickness and material are the same for all core shells (Appendix E.1).

Is there a label or target associated with each instance? If so, please provide a description. Yes. κ and the displacement field at the peak come from our simulations. The failure site and the site ratio q are derived from the field by one released rule (Appendix E.1). The reference shells carry κ only.

Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing. No. Every record is complete, and the list of defects of every shell is released. A model that reads the list of defects is a tabular model, reported separately (Section 4).

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit. Yes. Every sweep shell is a single-factor edit of a core shell and carries its own id and its source id. P1 to P3 share 100 sources, 99 of them sparse, all in the B1 test set. The 80 P4 sources fall in all three parts of each split (Appendix E.4). Sweep arms identical to their source are not in the sweep files.

Are there recommended data splits (e.g., training, development/validation, testing)? If so, please provide a description of these splits, explaining the rationale behind them. Yes, three fixed splits of the 4,399 core shells, drawn with seed 2027 (Appendix E.4, Table 6). B1 is random (3,299 / 440 / 660) and measures accuracy within distribution. B2 tests on deeper defects (2,789 / 376 / 1,234). B3 trains on sparse shells and tests on dense ones, where defects interact (1,934 / 265 / 2,200). The rationale is that B2 and B3 move the test set outside the training population in the two directions that change the collapse. The sweeps are in no split and are for evaluation only. They should not be used for training when scoring a split whose test set contains their sources.

Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description. The generation stored 4,999 core records, of which 600 duplicated another simulation. They were removed before any split was drawn, and none remain. The labels are checked (Appendix E.6, Table 10). κ is within 0.4% of a finer mesh on 5 of 6 study shells. The states one step before and after the peak show how sharply the displacements are defined. The site of near-ties ($q > 0.9$, 27.5% of the core) can change under a rotation of the defect pattern (Appendix E.5).

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? If it links to or relies on external resources, are there guarantees that they will exist and remain constant over time; are there official archival versions; are there any restrictions (e.g., licenses, fees) that apply to a future user? Self-contained. It links to no external resource. The experimental data used for validation (scans, pressure curves and photographs of Derveni et al. (2023)) are not redistributed.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)? If so, please provide a description. No.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? If so, please describe why. No. The dataset consists of numerical outputs of physics simulations and contains no human-generated content.

Does the dataset identify any subpopulations (e.g., by age, gender)? N/A. Instances are simulations of engineered shells, not human data.

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? If so, please describe how. No. The dataset contains no information about natural persons.

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? If so, please provide a description. No.

Any other comments? None.

N.3 Collection process

How was the data associated with each instance acquired? Was the data directly observable (e.g., raw text, movie ratings), reported by subjects (e.g., survey responses), or indirectly inferred/derived from other data (e.g., part-of-speech tags, model-based guesses for age or language)? If the data was reported by subjects or indirectly inferred/derived from other data, was the data validated/verified? If so, please describe how. Derived from physics simulation. Each shell was generated from its defect parameters on the release mesh and solved in Abaqus by arc-length continuation up to and past the first peak of the load (Appendix E.1). Verification by mesh refinement and by the states around the peak, and validation against the collapse pressure of eight scanned specimens, within 3.3% (Appendix E.6).

What mechanisms or procedures were used to collect the data (e.g., hardware apparatuses or sensors, manual human curation, software programs, software APIs)? How were these mechanisms or procedures validated? Abaqus/Standard (Dassault Systèmes Simulia Corp., 2024) on a CPU cluster, most solves on 8 cores. The displacements were kept at the peak and one step on either side of it, and written to one NumPy archive per shell with Python scripts. The procedure was validated as in the previous answer (Appendix E.6).

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)? Probabilistic for the core. Each shell draws its defects (11 to 18 for sparse shells, 69 to 81 for dense ones), their depths and their centres within 60° of the pole, under the minimum centre separation of its population, with the defect design of Derveni et al. (2023) (Appendix E.2). Deterministic for the sweeps, which edit chosen core shells one factor at a time (Appendix E.5).

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)? The authors only, compensated through their academic positions. No crowdworkers or paid annotators.

Over what timeframe was the data collected? Does this timeframe match the creation timeframe of the data associated with the instances (e.g., recent crawl of old news articles)? If not, please describe the timeframe in which the data associated with the instances was created. The core and the single- and two-defect shells were simulated between August 14 and September 14, 2026, the sweeps between September 21 and 24, 2026, and the eight specimens on September 23 and 24, 2026 (scheduler records, re-runs included). The released simulations took about 21,600 core-hours, with a median of 2.0 core-hours per core solve (Appendix A). Because the data is generated by simulation, the collection timeframe equals the creation timeframe.

Were any ethical review processes conducted (e.g., by an institutional review board)? If so, please provide a description of these review processes, including the outcomes, as well as a link or other access point to any supporting documentation. N/A. No human subjects, no animals, no personal data.

Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)? N/A.

Were the individuals in question notified about the data collection? If so, please describe (or show with screenshots or other information) how notice was provided, and provide a link or other access point to, or otherwise reproduce, the exact language of the notification itself. N/A. No individuals.

Did the individuals in question consent to the collection and use of their data? If so, please describe (or show with screenshots or other information) how consent was requested and provided, and provide a link or other access point to, or otherwise reproduce, the exact language to which the individuals consented. N/A.

If consent was obtained, were the consenting individuals provided with a mechanism to revoke their consent in the future or for certain uses? If so, please provide a description, as well as a link or other access point to the mechanism (if appropriate). N/A.

Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted? If so, please provide a description of this analysis, including the outcomes, as well as a link or other access point to any supporting documentation. N/A. No data subjects.

Any other comments? None.

N.4 Preprocessing / cleaning / labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? If so, please provide a description. If not, you may skip the remaining questions in this section. Yes. The 600 duplicated core records were removed. The displacement fields are stored as solved. The failure site and q are computed from them by the released rule. A spherical-grid view is produced on load by a map stored in the shared file (Appendix E.3).

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? If so, please provide a link or other access point to the "raw" data. The solver output databases are retained by the authors and not released. The released fields are the solver output at the stored steps, and the solver input of every record can be rebuilt from its list of defects and the settings of Appendix E.1.

Is the software that was used to preprocess/clean/label the data available? If so, please provide a link or other access point. Yes. The loader, the scorer and the code of every model and reference are released under the MIT licence, in the code repository referenced in Appendix B.

Any other comments? None.

N.5 Uses

Has the dataset been used for any tasks already? If so, please provide a description. This paper, which evaluates eight field models, tabular and set models on the list of defects, and reference baselines under the three protocols, the intervention tests and a few-shot study (Section 5).

What (other) tasks could the dataset be used for? (i) Surrogate models of the collapse pressure, the collapse field and the failure site from surface geometry. (ii) Sparse-to-dense transfer (B3) and extrapolation to deeper defects (B2). (iii) Few-shot adaptation. (iv) Predictors that report or use the site ratio between competing sites. (v) Tabular and set models on the list of defects, and their comparison with models that read the surface.

Is there a repository that links to any or all papers or systems that use the dataset? If so, please provide a link or other access point. It will be maintained on the project page and the code repository (anonymized links in Appendix B).

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? For example, is there anything that a future user might need to know to avoid uses that could result in unfair treatment of individu-

als or groups, or other risks or harms (e.g., legal risks, financial harms)? If so, please provide a description. Is there anything a future user could do to mitigate these risks or harms? Yes. All core shells share one slenderness, one defect width, inward defects of uniform wall thickness and a clamped rim. Physical shells carry outward features and thickness variation (Appendix E.6.3). The site label is defined at the peak load, before the collapse. Models trained on BLISS should not be expected to hold for other shells, defects or loadings without retraining and validation (Section L).

Are there tasks for which the dataset should not be used? If so, please provide a description. Yes. Certification or design of real structures without independent validation. The motion after collapse. Later peaks of the load. Shells or loadings outside the sampled design (Section L).

Any other comments? None.

N.6 DISTRIBUTION

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? If so, please provide a description. Yes, publicly, through a public dataset repository.

How will the dataset be distributed (e.g., tarball on website, API, GitHub)? Does the dataset have a digital object identifier (DOI)? As release 1.0, one NumPy archive per shell with fields, parametric tables, split files and a SHA-256 manifest (Table 5). A review sample is hosted at an anonymized link during review (Appendix B), and the full release, with its Croissant metadata record, is available to reviewers on request. A DOI will be issued with the public release in the final version.

When will the dataset be distributed? The review sample at the time of submission through the anonymized link, the full release to reviewers on request during review, and the full dataset publicly with the final version.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? If so, please describe this license and/or ToU, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms or ToU, as well as any fees associated with these restrictions. CC BY 4.0 for the dataset (`https://creativecommons.org/licenses/by/4.0/`) and MIT for the code (Appendix C). No fees apply.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any relevant licensing terms, as well as any fees associated with these restrictions. No. The release contains no third-party data (Appendix C). The simulations were run with the commercial solver Abaqus under a university research licence. The release holds only the numerical results, and no Abaqus software or input files. The experimental data of Derveni et al. (2023) are used for validation only and are not redistributed.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? If so, please describe these restrictions, and provide a link or other access point to, or otherwise reproduce, any supporting documentation. No.

Any other comments? None.

N.7 MAINTENANCE

Who will be supporting/hosting/maintaining the dataset? The authors (disclosed in the final version).

How can the owner/curator/manager of the dataset be contacted (e.g., email address)? Through the corresponding author (withheld for anonymous review), or by opening an issue on the code repository.

Is there an erratum? If so, please provide a link or other access point. No erratum at submission time. Errata, if any, will be tracked in the change log of the dataset repository.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? If so, please describe how often, by whom, and how updates will be communicated to users (e.g., mailing list, GitHub)? Yes. Published versions are immutable. Corrections and additions are issued by the authors as needed, with no fixed cadence, as new versions with a change log, communicated on the dataset repository and the code repository. The splits of version 1.0 stay frozen.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted)? If so, please describe these limits and explain how they will be enforced. N/A. No human-subjects data.

Will older versions of the dataset continue to be supported/hosted/maintained? If so, please describe how. If not, please describe how its obsolescence will be communicated to users. Yes. Each version keeps its files and hashes, so the version used in this paper stays available and work that pins a release stays reproducible.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? If so, please provide a description. Will these contributions be validated/verified? If so, please describe how. If not, why not? Yes. A new shell is a new list of defects solved with the settings of Appendix E.1. Contributions are accepted as new versions, not as changes to an existing one, after the same checks (Appendix E.6).

Any other comments? None.

O LLM USAGE STATEMENT

In this work, we used generative AI tools (Claude and ChatGPT) to implement methods, namely the data-processing, training, evaluation and scoring code, to provide feedback on the research methodology and on the design of experiments, such as the intervention tests, the few-shot protocol and the resolution controls, to support data analysis and the interpretation of results, to refine the framing of the hypotheses, to clean and reformat the dataset for release, and to assist with translation from Portuguese. We have not used generative AI tools to generate data. Every shell is a finite-element simulation run by the authors, and the tools only helped write the scripts that build, run and post-process the solver input. Formulating mathematical claims and writing proofs are not applicable to this work. Additionally, we used generative AI tools to create and modify figures, to create and edit code, to draft and edit text for readability, to suggest the paper structure, title and keywords, and to identify, summarise and check related work and format references. We have reviewed all AI-assisted work. Every number in the paper was re-derived from the prediction files by the released scorer and scripts, reference metadata were checked against publisher records, and research decisions, protocols and claims were made by the authors. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.