Neural Surface Potential: Supplementary Material

Marcel Padilla
ETH Zürich
Zürich, Switzerland
marcel.padilla@inf.ethz.ch

Abstract

This document accompanies the paper *Neural Surface Potential*. It holds the data pipeline, the training step and recipe, the network, the metric definitions, the label-accuracy study, the full accuracy and orientation tables, the training runs, the data-scaling measurement, the approaches that did not work, and further qualitative results. Section, figure, table and equation numbers without an S refer to the paper.

S1 Data

Fig. S1 summarizes the pipeline of Sec. 5. The two sources fail in different ways. Thingi10K [Zhou and Jacobson 2016] contains mechanical, 3D-printable parts with complex topology. Manifold40 [Hu et al. 2022] contains smoother household and vehicle categories, many of them close to bodies of revolution. Manifold40 is the watertight remesh of ModelNet40 [Wu et al. 2015]; we use it because the exterior Neumann problem needs a closed surface and the original meshes are not closed. Its meshes have exactly 500 faces, so our remeshing upsamples them by about 24×. Because the 403-mesh validation set is 87% Manifold40, pooled validation statistics are dominated by the easier source, and the test set is therefore Thingi10K only.

Splits and leakage. We split by design family (583 families), because user-uploaded collections contain near-duplicate variants of one object that a per-mesh split would place on both sides. The validation and test sets were frozen before any run reported in the paper. Three screens were applied to the training set: against the validation file names, against the test file names, and a category hold-out that removed a further 226 meshes. A name-collision screen catches the same object uploaded under different identifiers; without it, 9 of 54 meshes in one draw leaked across the split. Mesh counts overstate diversity: in one training subset a quarter of the meshes are near-exact copies of a sibling, separated when a multi-part file was split into components.

S2 Training

Fig. S2 shows how rotation augmentation obtains exact labels (Sec. 4.2).

S2.1 The recipe in full

AdamW [Loshchilov and Hutter 2019] with weight decay 0.01, batch 20, peak learning rate 2.0×10^{-3} annealed by cosine to 2.0×10^{-5}, 4% of the schedule as linear warmup, point dropout 0.95 (each sample keeps a uniformly random 5 to 100% of its vertices), exact SO(3) rotation augmentation, an exponential moving average of the weights with decay 0.999, and no gradient clipping. The loss is (8) with the denominator floored at 0.1 times the mesh's joint potential scale. The final model runs 1050 epochs, so its warmup is 42 epochs.

The recipe gain of Sec. 4.4 (13.62 → 9.65%) came from raising the batch size from 10 to 20 with the learning rate scaled by $\sqrt{2}$ and adding an 80-epoch linear warmup, 4% of those 2,000-epoch runs. Three settings were chosen over alternatives that performed worse. Weight decay 0.05, the value PointNeXt uses for its own task [Qian et al. 2022], is worse here than no decay at all. Gradient clipping at 1.0 hurts multi-mode runs, although it is required when training a single rotational channel, which otherwise collapses to the zero-field predictor (the loss reaches exactly 1.0 and stays there). A cyclical schedule designed to benefit the weight average is 0.63 points worse than a single cosine anneal (Table S5). The weight average makes checkpoint selection reliable but does not improve the final accuracy by itself.

S2.2 The output filter

Plain uniform diffusion, which shrinks the signal, is 1.75 points worse than Taubin smoothing at $k = 8$. Training through the filter costs 0.76 points and more for stronger filters. The network then learns to pre-distort: its unfiltered output is about a point worse than that of the plain network, and after filtering it reaches 5.88% against 5.42% for the filtered plain network. These two numbers and the roughness ratio of Sec. 4.5 come from the 2,000-mesh configuration of Table S5.

S3 The network

Fig. S3 shows the encoder-decoder at the sizes of Sec. 4.3. The three-layer per-vertex head maps the concatenation of the finest local feature and the global pooled feature to two channels. The ball radii are absolute lengths in the canonical frame. The neighbourhood sizes add no parameters but change what the network computes, so the model must be rebuilt from its stored configuration and never by hand; a shared neighbourhood cap of 32 had silently collapsed the sizes in earlier runs (Sec. 4.4).

S4 Metrics

Denominators. The paper reports the per-mode relative error of Sec. 6.1, each mode divided by its own norm. Our training logs and the released run directory also contain a *joint* relative error, with one denominator per mesh: the root-mean-square reference norm over the channels the model predicts. On our data the two agree to within a few percent on the translational modes, but the joint metric flatters ϕ_{rx} and ϕ_{ry} by 3.0 to 3.2× and ϕ_{rz} by 16.1×. It is also not comparable across models with different numbers of output channels, because its denominator depends on them: between our two-channel and six-channel configurations the median denominator ratio is 0.749. Cross-configuration comparisons in the paper therefore use the per-mode metric, and the joint metric appears only in Tables S5

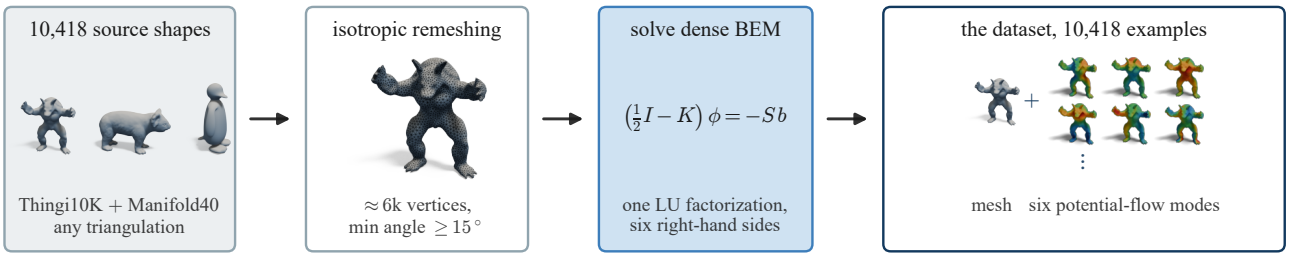

Figure S1: Training data. Every shape is remeshed isotropically to about 6,000 vertices with no triangle angle below 15°, because the labels come from a constant-panel collocation solve whose error is set by the worst triangles. One factorization serves all six Kirchhoff modes. A training example is a mesh with its six solved modes; there are 10,418 of them. The bodies drawn are illustrations and are not part of the data.

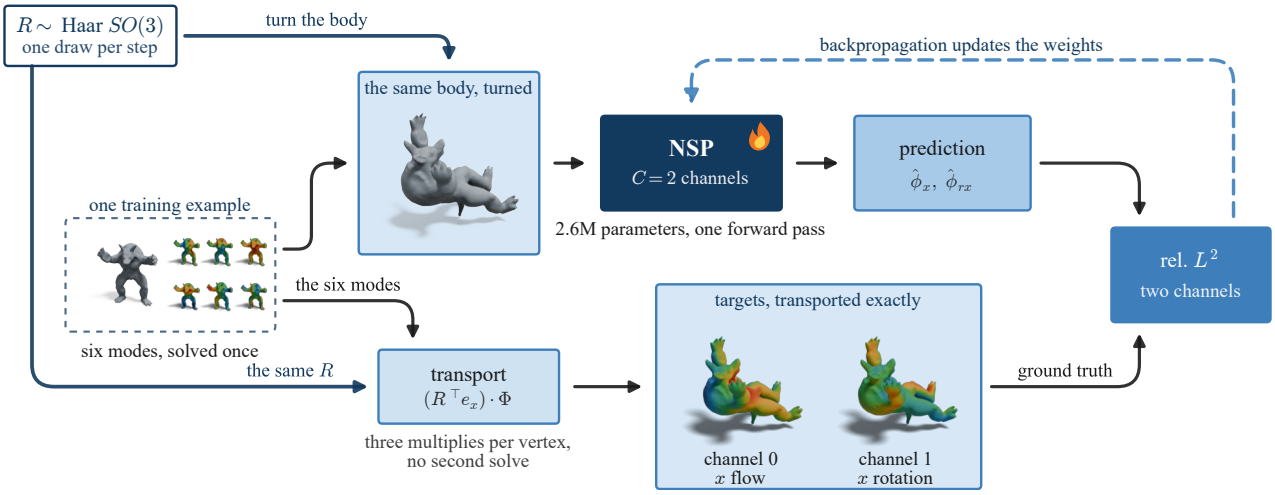

Figure S2: One training step. A Haar-uniform rotation R turns the body, and the targets for the rotated copy are *transported* by (7) instead of solved again: each is a linear combination of three stored fields, so every rotation of a solved shape is a new example with exact labels. The loss compares the prediction with the transported targets. Only the x pair is supervised; three passes recover the other four modes at inference (Fig. 3). The flame marks weights being trained.

and S6, where all rows predict the same two channels. The training objective uses a third variant, the per-mode metric with a floored denominator (Sec. 4.4), which is never reported as a result.

The degeneracy in numbers. On the final model, one of the 403 validation meshes falls below 3% of the set median reference norm on ϕ_{rx}; with it the ϕ_{rx} mean is 9.18%, without it 6.89%, and the median moves by 0.04 points. A looser cut, dropping the ten meshes whose rotational reference norm is below 0.02, gives a ϕ_{rx} mean of 6.80% and lowers the worst case from 929% to 42.6%. Every mesh with a well-defined denominator stays in, including the failure cases of Fig. 4.

S5 Label accuracy

Table S1 is the convergence study behind the label accuracy quoted in Sec. 5. Each row halves the panel size. The face error converges

Table S1: The label floor: BEM error against the analytic unit sphere ($\phi|_S = -(R/2)\, n_x$, $\mathsf{M}_{xx} = 2\pi/3$) at increasing panel counts, in the paper's gauge-fixed, area-weighted relative L^2. The vertex column, after the face-to-vertex scatter, is what the network is trained on. Training labels are solved at roughly 12,000 faces.

Faces	Vertices	face rel-L^2 (%)	vertex rel-L^2 (%)	M_{xx} (%)
80	42	1.575	6.557	7.826
320	162	1.114	2.485	2.202
1,280	642	0.639	0.939	0.656
5,120	2,562	0.341	0.351	0.219

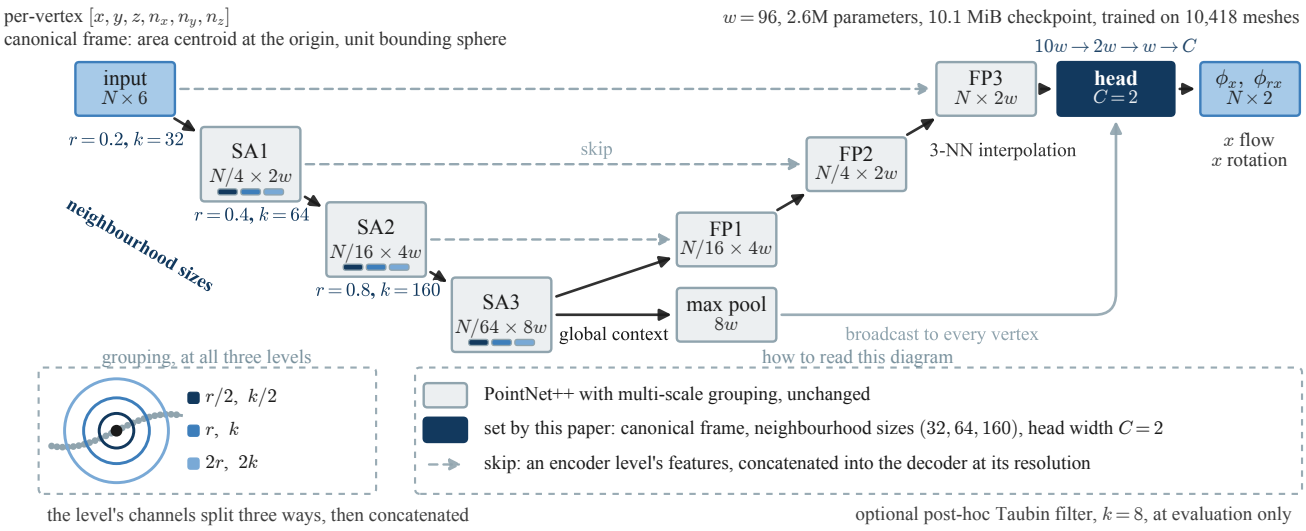

Figure S3: The PointNet++ backbone and our choices. The standard backbone is drawn in outline: an encoder-decoder with multi-scale grouping at all three set-abstraction levels, feature propagation by three-nearest-neighbour interpolation, one global max pool, and skip connections. Filled blue marks what we set: the canonical frame and the six per-vertex input channels, the neighbourhood sizes $(32, 64, 160)$**, and the head width** $C = 2$**. Multi-scale grouping splits a level's channels three ways and adds only** 4.9% **parameters, for about** 2.6M **in a** 10.1 **MiB checkpoint. All shapes were read from a built model;** N **is the input vertex count.**

at an order rising towards one, as expected for piecewise-constant collocation; the vertex error, which averages neighbouring panels, converges at order 1.4, and the added-mass entry at 1.6 to 1.8, the same effect of integration that the network's tensor errors show. The sphere also checks conventions: its rotational block vanishes by symmetry, and recovering this together with a positive definite translational block confirms that solver, labels and metric agree on normal orientation and reference point. The sphere is the easiest case for constant panels; on thin plates and deep cavities, where the network is also least accurate, the label error was not measured.

S6 Accuracy in full

Table S2 adds three columns to Table 1: the mean error over all meshes, including the degenerate ones, the median absolute error, and the median norm of the exact field. It also adds a block with the validation meshes from Thingi10K, the only source of the test set. On the validation set the rotational potentials are about three times weaker than the translational ones (last column), so our figures never draw both families on one scale. Pooled over the three modes of each family, translational and rotational medians differ by a factor of 1.28 and their 75th percentiles by 1.23. For the trained pair, ϕ_x and ϕ_{rx}, the medians differ by a factor of 1.06.

S7 Orientation in full

Table S3 and Fig. S4 are the measurements behind Sec. 6.5. The canonical column of the table reproduces Table 1. Positions, normals and both label triples rotate together.

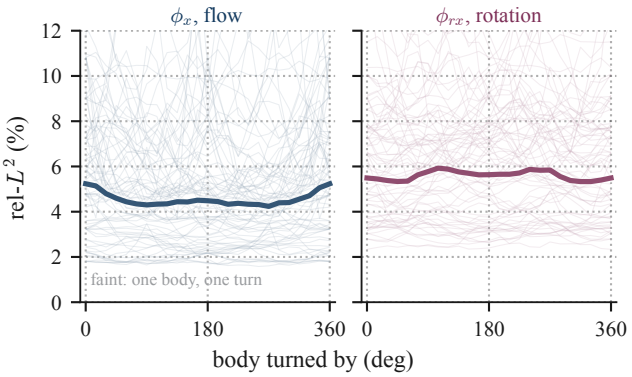

Figure S4: Rotating an unseen body does not degrade the prediction. Each faint trace is one validation mesh turned through a full revolution about a random axis; the heavy curve is the median over 405 **turns (**135 **meshes,** 3 **axes,** 25 **angles), of which** 80 **are drawn. Every trace returns to its starting value at** 360°**, which checks the evaluation. The medians vary over** 4.2–5.2% **and** 5.3–5.9%**, partly because the metric's denominator rotates with the label. The axis is cut at** 12%**, above which lie** 9.2% **of the samples.**

Why the relative error can fall. Rotating the body also rotates the label, and with it the denominator of the metric. A random mixture of the three translational modes has 1.30× the norm of the canonical ϕ_x, while the absolute error grows by only 1.12×. For the rotational

Table S2: Main result in full (Table 1 is the compact version). All six Kirchhoff modes and the complete 6×6 added-mass tensor from one two-channel network. Only the two modes marked † have output channels; the other four come from the same weights on cyclically rotated copies of the mesh (Sec. 4.2) and are an independent measurement. Field error is the gauge-fixed, area-weighted relative L^2 against the exact BEM solution, each mode against its own norm; Taubin $k = 8$ is the post-hoc filter of Sec. 4.5, without retraining. The last two columns, the median *absolute* error and the median norm of the true field in the canonical frame, put the relative columns in scale. $*$ marks means over a set that contains meshes on the structural $0/0$ of Sec. 6.1. The third block is the part of the validation set that comes from Thingi10K, the only source of the test set. The network is equivariant only approximately: the six modes it predicts for a randomly rotated copy of a mesh, rotated back, differ from those for the mesh itself by 4.6% at the median and 9.5% at the 90th percentile over the 403 validation meshes.

	Field rel-L^2 (%), raw		Field rel-L^2 (%), Taubin k=8		Added-mass	abs. err	field
Mode	median	mean	median	mean	row error (%)	median	$\|\|\phi\|\|$
Validation set, 403 meshes							
ϕ_x †	5.19	5.96	4.45	5.19	1.15	0.0118	0.282
ϕ_y	4.72	6.01	4.01	5.29	1.18	0.0109	0.270
ϕ_z	4.02	5.98	3.40	5.36	1.01	0.0147	0.441
ϕ_{rx} †	5.48	9.18*	4.55	7.98*	2.16	0.0057	0.125
ϕ_{ry}	5.80	6.97	4.79	6.00	2.65	0.0047	0.098
ϕ_{rz}	6.73	28.53*	5.57	25.13*	2.91	0.0040	0.064
tensor: translation / rotation block					0.91 / 0.82		
whole tensor, Frobenius					**1.05** (p90 3.00)		
Validation set, Thingi10K meshes only, 50 meshes							
ϕ_x †	6.62	7.51	5.74	6.48	1.52	0.0141	0.278
ϕ_y	6.14	7.02	5.22	6.06	1.45	0.0168	0.282
ϕ_z	5.51	6.95	4.90	6.11	1.69	0.0204	0.429
ϕ_{rx} †	7.65	26.80*	6.64	23.22*	2.86	0.0069	0.116
ϕ_{ry}	7.18	8.40	5.96	7.23	2.89	0.0086	0.127
ϕ_{rz}	10.32	89.01*	8.69	83.68*	3.77	0.0052	0.053
tensor: translation / rotation block					1.26 / 1.57		
whole tensor, Frobenius					**1.39** (p90 3.24)		
Held-out test set, 77 meshes							
ϕ_x †	6.40	8.17	5.20	7.28	1.66	0.0158	0.297
ϕ_y	5.81	8.27	4.79	7.40	1.43	0.0153	0.299
ϕ_z	5.31	8.76	4.40	7.81	1.64	0.0207	0.457
ϕ_{rx} †	7.28	9.41	6.06	8.37	2.81	0.0082	0.136
ϕ_{ry}	7.00	9.79	6.06	8.70	3.24	0.0074	0.116
ϕ_{rz}	10.16	939.07*	9.25	885.70*	4.44	0.0056	0.063
tensor: translation / rotation block					1.46 / 1.47		
whole tensor, Frobenius					**1.58** (p90 5.84)		

triple the norm ratio is 0.85×, and the absolute error falls with it. Conversely, ϕ_x is the smallest of the three translational modes on 42% of these meshes and the largest on 15%, so the canonical ϕ_x error is measured against a systematically small denominator and is conservative.

Augmentation. The control pair was trained on 83 meshes and is scored on the 32-mesh set it was selected on, where both arms are competent; on the 403-mesh validation set both approach the error of a constant predictor and the contrast saturates (table note). Augmentation also lowers the canonical error by a factor of 1.7 in this pair. Because the transported labels are exact, each rotated example is a new training sample, so augmentation acts as a data multiplier as well as a symmetry prior. This agrees with the artery study [Suk et al. 2024], in which the same backbone collapses under rotation without augmentation and recovers with it.

S8 Resolution on one body

In Fig. 6 of the paper, predictions on decimated meshes are scored against the tensor of the full-resolution body, so the error contains a decimation term. In Fig. S5 each row is solved exactly at its own resolution, so the difference is the network's error alone. At 2,302 vertices the error is 1.7× that at the training resolution.

S9 Backbone screen

Table S4 is the screen summarized in Sec. 6.7. All four backbones need no per-mesh precomputation. The transformer underfit at this schedule (its training error stayed far above that of PointNet++ at a

Table S3: Orientation robustness. Each model is evaluated in the meshes' canonical orientation and under 8 Haar-uniform random rotations per mesh, against a rotated ground truth that is exact and free by (7). The two controls differ in one training flag, rotation augmentation, and are a small early run, so read their ratios, not their level. *Rel.* is the median over meshes of each mesh's rotated-to-canonical relative error, *Abs.* the same ratio unnormalised; they differ because the label rotates too, the transported ϕ_x carrying 1.30× the norm of the canonical one on this data.

Model	Test set	Mode	Canonical (%)	Rotated (%)	Rel.	Abs.
NSP (final)	validation, 403	ϕ_x	5.19	4.27	0.92×	1.12×
NSP (final)	validation, 403	ϕ_{rx}	5.48	5.49	1.03×	0.89×
NSP (final)	test, 77	ϕ_x	6.40	5.33	1.00×	1.24×
NSP (final)	test, 77	ϕ_{rx}	7.28	7.20	1.01×	0.89×
Control, augmented, seed 0	selection set, 32	ϕ_x	15.64	15.73	1.06×	1.11×
Control, augmented, seed 1	selection set, 32	ϕ_x	14.74	15.43	1.04×	1.14×
Control, no augmentation, seed 0	selection set, 32	ϕ_x	24.40	34.52	1.26×	1.47×
Control, no augmentation, seed 1	selection set, 32	ϕ_x	26.94	33.64	1.21×	1.33×

The controls are scored on the 32-mesh set their checkpoints were selected on, because a ratio is only meaningful where the model is competent. On the 403-mesh validation set, which is far outside their 83-mesh training distribution, both sit near the error a constant predictor would score and the contrast saturates: aug 16.3/18.6 (1.10×); aug 16.2/19.4 (1.16×); no-aug 37.5/42.7 (1.14×); no-aug 37.5/42.7 (1.12×). The final model's own numbers are unaffected, since this is its validation set.

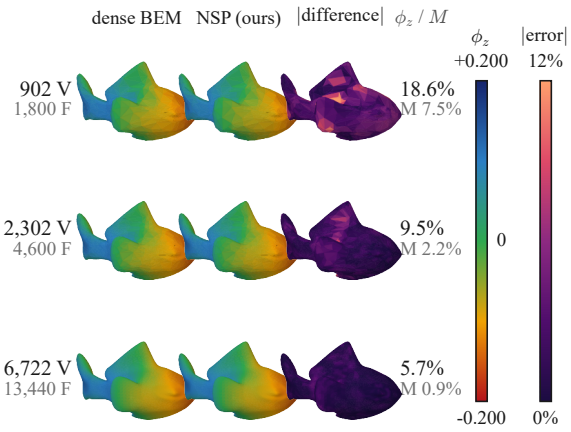

Figure S5: One unseen body, the fish, solved exactly at three nested decimations of one surface: the exact solve, ours, and the magnitude of their difference, on one potential scale and one error scale capped at the 99th percentile. The prediction remains close to the solve down to 2,302 vertices (9.5% relative L^2 in ϕ_z, the mode along the body's length, against 5.7% at 6,722) and degrades below, to 18.6% at 902 vertices. The tensor error is 0.9% at the finest and 7.5% at the coarsest level.

similar parameter count), and a longer run with warmup improved it only to 33.5%.

S10 Ablations

Table S5 is the ablation sequence summarized in Sec. 6.7: from a two-channel baseline of width 64 to the final configuration, plus single changes that were measured and not adopted. None of the adopted changes alters the model family. The width sequence saturates:

Table S4: Real-time backbone screen: four feed-forward architectures without per-mesh precomputation, trained identically (300 meshes, 300 epochs, one seed, RTX 3090, a recipe two generations older than the final one). Validation error is the lowest reached; every arm was still improving when it stopped, so these are viability numbers, not a converged comparison. The last row is the transformer at four times the schedule, with warmup.

Backbone	Params (M)	Epochs	Best val rel-L^2 (%)	Train (min)
PointNet++, NSP (ours)	1.12	300	22.39	14
PVCNN	0.94	300	25.46	35
Transolver	0.98	300	35.94	51
Graph U-Net	1.34	300	36.98	38
Transolver, 4× schedule	0.98	1200	33.54	181

width 128 adds about half as much again as width 96 gained, and width 192 adds only 0.11 points more for four times the parameters of width 96.

S11 Training runs and convergence

The final model trains for 1050 epochs over 10,418 meshes in 87 h 45 min on one NVIDIA Quadro RTX 6000. Table S6 lists the three configurations trained at this scale with equal wall-clock budgets, and Fig. S6 shows their convergence.

We sized the three runs with a measured per-visit cost model (1.54× for multi-scale grouping, 1.27× for the wider model) and recorded a prediction before launch: at the measured −0.51 points per doubling of the schedule, grouping should gain about 0.38 points and width only 0.06. Both predictions held in sign and rank. Grouping gained 0.41 points and width 0.17, at realized per-epoch costs of 1.63× and 1.34×, so both kept most of their matched-epoch gains (0.41 of 0.54 and 0.17 of 0.22; Table S5). Validation error is flat over the final tenth of the schedule, at the learning-rate floor, and

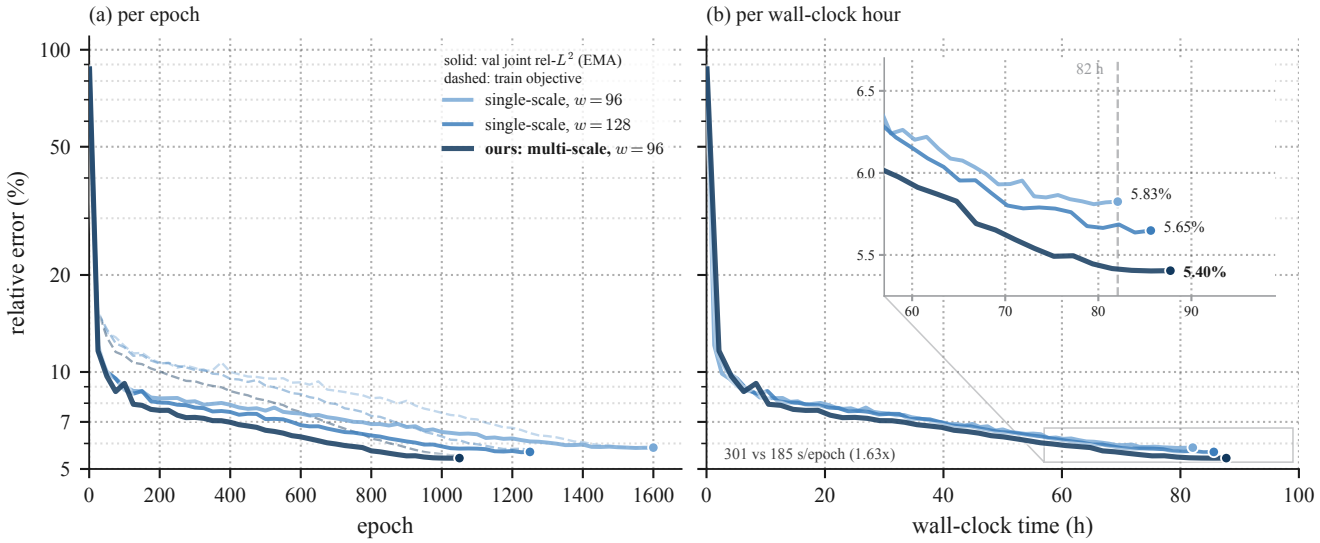

Figure S6: Training curves of the three runs of Table S6: validation error against epoch (left) and against wall-clock time (right). The epoch counts were chosen to equalize compute, so the right panel is the fair comparison. The annotated values are those of the last epoch; Table S6 reports the best EMA epoch.

Table S5: Ablations at matched epochs: every arm runs 2000 epochs with seed 2 on the same 2,000 training and 403 validation meshes and differs from its reference in one setting. The metric is the joint training metric (Sec. S4 of the supplement), comparable across rows but not with the per-mode errors elsewhere. Negative deltas are improvements; single seed.

Configuration	Params (M)	Joint EMA (%)	Δ (pts)
two-channel baseline, width 64	1.12	7.30	n/a
+ loss-denominator floor	1.12	7.03	-0.26
+ width 96	2.52	6.62	-0.41
+ multi-scale grouping	2.65	6.08	-0.54
single changes measured against the width-96 row, none adopted			
width 128 instead of 96	4.48	6.39	-0.22
width 192 instead of 96	10.08	6.29	-0.33
+ Sobolev (H^1) loss term	2.52	6.45	-0.17
denser SA centroids	2.52	6.67	+0.05
train through the output filter	2.52	7.38	+0.76
measured against the width-64 baseline			
cyclic schedule + weight averaging	1.12	7.93	+0.63
difficulty-weighted mesh sampling	1.12	7.24	-0.05

the last 50 epochs change it by 0.01 points. A second seed of the final configuration would cost about 88 GPU-hours and was not run.

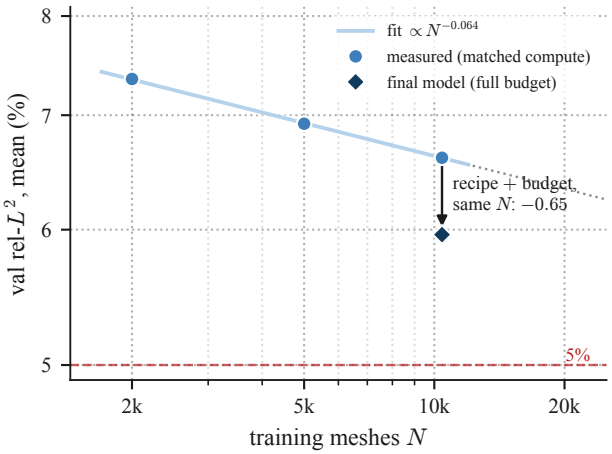

Figure S7: Mean ϕ_x validation error against training-set size at a fixed budget of 4.0M mesh visits, so every point has equal compute. The sets are nested subsets of the full labelled pool, so no larger set exists. The fitted exponent is −0.064; the dotted line continues the fit and the dashed line marks 5%. The diamond, the final model at the largest size, is not in the fit because it changes both compute and recipe, which together gained 0.65 points at the same N, against 0.74 for the whole fivefold increase in data. The paper's tables report medians.

S12 Data scaling

Fig. S7 is the measurement behind the data-scaling paragraph of Sec. 6.7. The error falls slowly with data and has not flattened at the largest size we could label. Three points over a 5.2× range do not determine the exponent precisely. The figure continues the fit only to twice the largest set, and the paper's estimate for a 5% error, a hundredfold more data, is an order of magnitude, not a prediction.

Table S6: The three final runs: the same 10,418 training meshes, the same 403-mesh validation set and the same wall-clock budget, hence different epoch counts. The joint column is the training metric (one denominator shared by the two predicted channels) at the best EMA epoch; the last two columns are the per-mode field errors reported elsewhere in the paper, which rank the runs identically. The final model is in bold.

					Field rel-L^2 (%), median	
Configuration	Epochs	Wall (h)	Params (M)	Joint EMA (%)	ϕ_x	ϕ_{rx}
multi-scale grouping, width 96	1050	87.7	2.65	5.40	5.19	5.48
single-scale grouping, width 96	1600	82.1	2.52	5.81	5.30	5.64
single-scale grouping, width 128	1250	85.6	4.48	5.64	5.31	5.58

S13 What did not work

A physics-informed loss. The trace of a function that is harmonic in the *exterior* is not harmonic on the surface, so penalizing $\Delta_S\phi$ leaves a nonzero residual at the true solution; we confirmed this on the analytic sphere. No purely local correction can work either, because the Neumann-to-Dirichlet map is nonlocal.

A principal-part decomposition. Splitting off an analytic principal part and learning the remainder works for the Dirichlet-to-Neumann direction [Ling et al. 2026]. For us, evaluating $(-\Delta_S)^{-1/2}$ costs about three times our entire six-mode inference, per mode. It would also help little: their operator has order +1 and a dominant principal part, whereas ours has order −1 and a residual rougher than the field it corrects. As a by-product, fitting the Dirichlet-to-Neumann operator as $\Lambda \approx |c|\sqrt{-\Delta_S}$ gives $|c| = 1$ to within 11% on all six modes, which confirms the symbol of the operator numerically.

Architecture and loss variants. Dilated grouping at its source paper's best setting ($d = 1, 4, 8$) is 0.76 points worse, consistent with that paper's caveat that a receptive field growing too fast starves the early layers. A per-vertex thin-wall descriptor helps, but a shuffled control that destroys its spatial assignment while keeping its per-mesh statistics comes within 0.09 points, so the benefit comes from a global summary. An H^1 seminorm term gives −0.17 points, within our single-seed resolution, and denser set-abstraction centroids give +0.05 points at 1.30× the forward-pass cost, an overhead that grows with mesh size (Table S5).

S14 More qualitative results

Figs. S8 and S9 extend Fig. 4.

S15 Reproduction

Everything is implemented in PyTorch [Paszke et al. 2019]. The farthest-point sampling and radius queries of every set-abstraction level use the kernels of PyTorch Geometric [Fey and Lenssen 2019], the only non-standard dependency of the forward pass. The released archive contains the weights of the final model (2,648,738 parameters in 10.1 MiB, file `checkpoint_best_ema.pt`), its run configuration and its per-epoch training log. With the code and the evaluation meshes, which the split lists rebuild, they reproduce every result of the paper except the backbone screen, the ablations, the training-run comparison and the data-scaling points, which come from separate runs. The model must be built from the stored configuration, because the neighbourhood sizes add no parameters but change what the network computes.

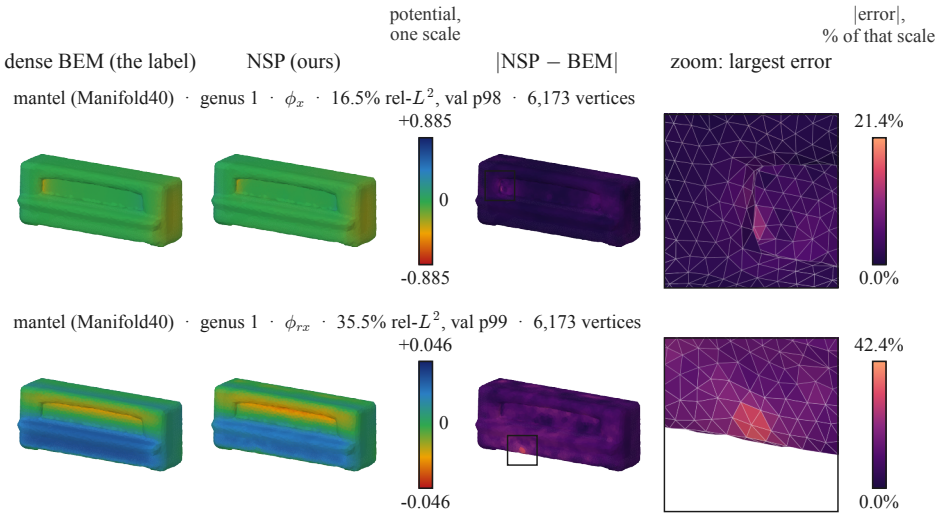

Figure S8: A difficult case in both mode families: a genus-1 mantel, among the least accurate few meshes of the validation set in each, and one of the deep-cavity shapes of Sec. 7. Only the mode changes between the rows; camera, columns, scales and zoom follow Fig. 4. The prediction still matches the label by eye, and the error is a broad, smooth offset without a localized break.

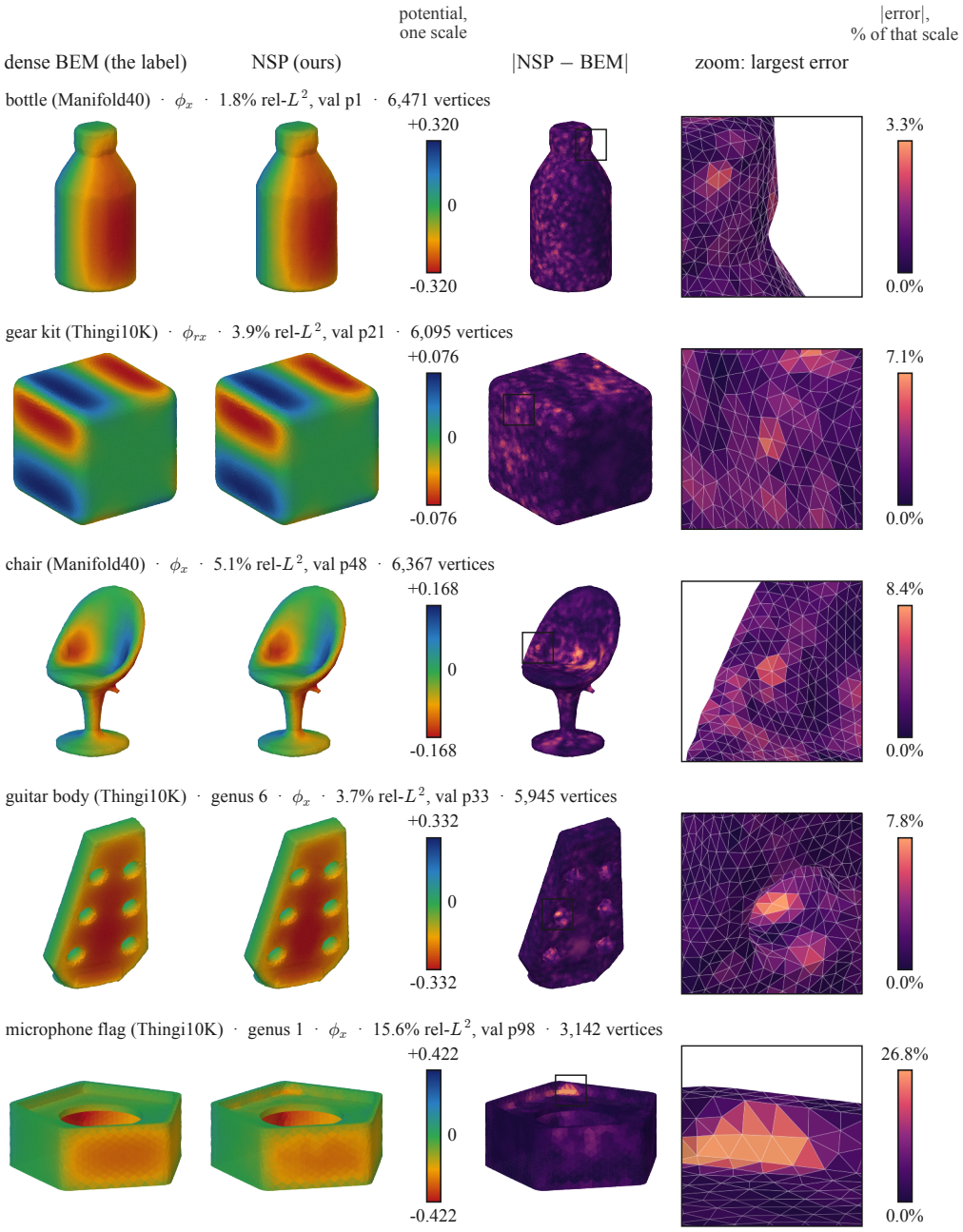

Figure S9: Five more held-out shapes from both sources and both mode families, spanning validation percentiles from the first to the last few percent (given in each row header). Columns, scales and zoom follow Fig. 4. Two rows lie near the accurate end of the validation distribution, two near its median and one in its last two percent, a thin-walled hollow box.

References

Matthias Fey and Jan Eric Lenssen. 2019. Fast Graph Representation Learning with PyTorch Geometric. arXiv:1903.02428 [cs.LG] https://arxiv.org/abs/1903.02428

Shi-Min Hu, Zheng-Ning Liu, Meng-Hao Guo, Jun-Xiong Cai, Jiahui Huang, Tai-Jiang Mu, and Ralph R. Martin. 2022. Subdivision-based Mesh Convolution Networks. *ACM Transactions on Graphics* 41, 3 (March 2022), 1–16. doi:10.1145/3506694

Shuo Ling, Wenjun Ying, and Han Zhou. 2026. Principal-Part Decomposition for Neural Operator Learning of Dirichlet-to-Neumann Maps. arXiv:2606.25952 [math.NA] https://arxiv.org/abs/2606.25952

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101 [cs.LG] https://arxiv.org/abs/1711.05101

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG] https://arxiv.org/abs/1912.01703

Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. 2022. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 23192–23204. doi:10.52202/068431-1685

Julian Suk, Pim de Haan, Phillip Lippe, Christoph Brune, and Jelmer M. Wolterink. 2024. Mesh Neural Networks for SE(3)-Equivariant Hemodynamics Estimation on the Artery Wall. arXiv:2212.05023 [cs.LG] doi:10.1016/j.compbiomed.2024.108328

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. arXiv:1406.5670 [cs.CV] https://arxiv.org/abs/1406.5670

Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. arXiv:1605.04797 [cs.GR] https://arxiv.org/abs/1605.04797