# FLOATBENCH: A TABULAR DATASET AND BENCHMARK FOR FATIGUE PREDICTION ON FLOATING OFFSHORE WIND TURBINE TOWERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Tabular surrogates are commonly ranked on random validation splits. Yet a surrogate is used to predict wherever no simulation exists, both between training points (interpolation) and beyond the training domain (extrapolation), and random splits cannot tell the two apart, so the selected model may fail where it extrapolates. We introduce **FLOATBench**, a tabular dataset and benchmark for fatigue prediction on floating offshore wind turbine towers that tests whether model selection stays valid as predictions move from interpolation to extrapolation. It provides $582{,}120$ per-section fatigue-damage labels from $19{,}404$ OpenFAST simulations of three 22 MW tower geometries over $1{,}078$ wind–wave operating conditions. FLOATBench is distinguished by a regime-aware partition that places each test point relative to the training domain as in-train, interpolation, or extrapolation, so extrapolation depth is controlled by construction, and by cross-tower transfer with condition-level bootstrap intervals. Across $735$ trained surrogates and seven standalone baselines on three towers, the global rank-1 model falls to ranks 23, 11, and 11 in joint wind-and-wave extrapolation, where a network ranked 69th–79th globally wins (a rank reversal), and validation-based selection leaves $1.5$–$1.9\times$ the best available extrapolation error. In cross-tower transfer, the relative $L^2$ error is $0.067$–$0.098$ when the reference tower is in training versus $0.423$ when it is held out. The rank reversal holds across all 15 tested partition settings and after retraining on two alternative training grids (REF). Limitations: one 22 MW design family and aligned power-production conditions. Resources: dataset and code at `https://anonymous.4open.science/w/FLOATBench-115C/index.html`.

## 1 INTRODUCTION

Tabular surrogates are commonly ranked on random validation splits, but in use they are also queried near and beyond the edges of the sampled operating envelope, where a random split, filled with interior points, hides extrapolation failures. Floating offshore wind turbine (FOWT) tower fatigue is a clear instance of this setting. As shallow-water sites become limited, offshore wind is moving to deep water, reachable only by floating designs (Alves Ribeiro et al., 2025), and at 22 MW the tower becomes a fatigue bottleneck under coupled aerodynamic, hydrodynamic, mooring, and gravity loads. Fatigue must be checked over the whole wind–wave operating envelope, but OpenFAST (Jonkman, 2013) costs roughly one CPU-hour per 10-minute condition and a design iteration crosses thousands of conditions, so surrogates are trained on a sampled grid of conditions and then queried across, and beyond, it. Surrogates are standard for such bottlenecks in scientific ML (Elrefaie et al., 2024; 2025; Takamoto et al., 2022), yet FOWT fatigue has no shared benchmark that tests where they fail.

Existing wind-fatigue datasets and works mostly cover a single tower geometry, sample sparse or site-specific wind/wave operating points, provide labels for at most two tower cross-sections, and evaluate surrogates under random train/test splits, each with its own simulations, split, and metrics. The gap is more than a missing dataset: a benchmark for engineering surrogates must test reliability in extrapolation beyond the sampled envelope, not just interpolation inside it.

For FOWT fatigue, the deployment-relevant coordinates are physical and interpretable (mean wind speed, wind-speed standard deviation as a turbulence-intensity proxy, significant wave height $H_s$,



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.

peak wave period $T_p$), so the operating envelope admits a clean geometric stratification of test points into In-train, Interpolation, and Extrapolation; a random partition leaves the extrapolation cells empty.

**FLOATBench** closes this gap with a public, multi-geometry, per-section fatigue benchmark paired with an evaluation protocol matched to the operating envelope (Figure 1). Our contributions are: **(i) Dataset:** 582,120 per-section fatigue labels from 19,404 OpenFAST simulations (6,468 per tower) across three 22 MW FOWT towers, 1,078 aligned wind/wave states, six turbulence seeds, and 30 tower sections. **(ii) Protocol:** a regime-aware alpha-shape partition of the operating envelope covering all nine wind/wave regime cells, with per-section reliability and a reproducible evaluation harness. **(iii) Benchmark:** a test of whether model selection stays valid as deployment moves from interpolation to controlled extrapolation, not of which model has the lowest average error. It exposes rank shifts between global and subgroup-specific performance (regimes, sections, cross-tower transfer) invisible to random splits.

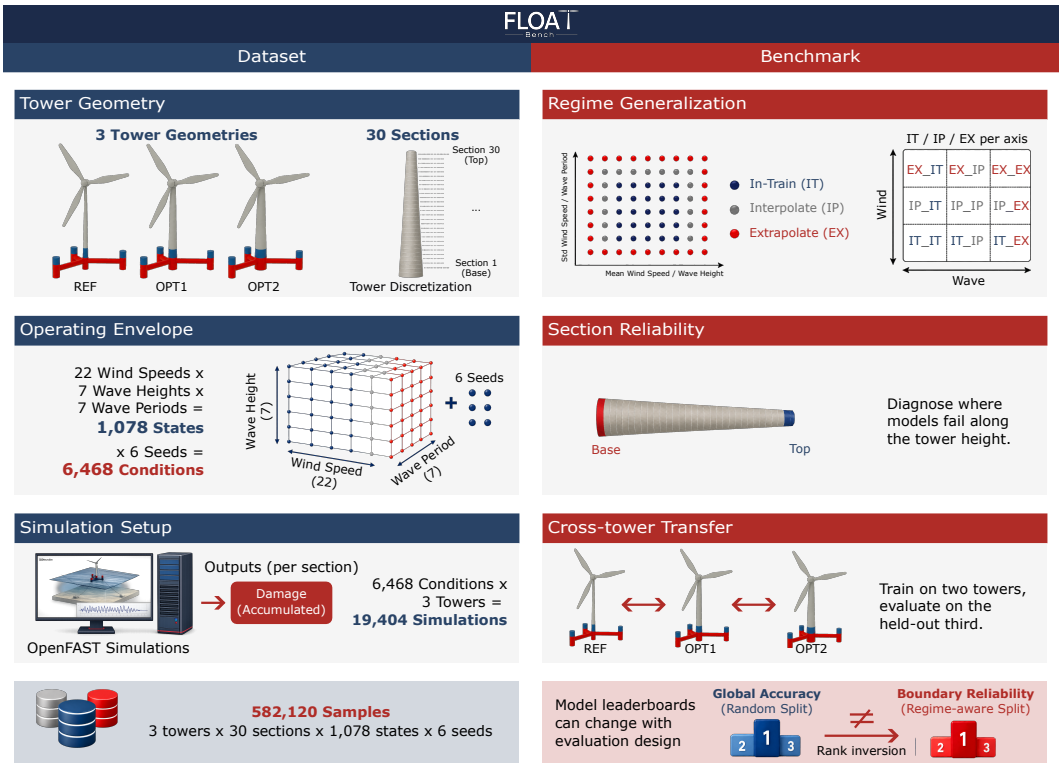



Figure 1: **FLOATBench:** a dataset and benchmark for 22 MW FOWT tower fatigue. Dataset: three 22-MW floating tower geometries, 1,078 aligned wind and wave states, 6 seeds, and 30 sections yield damage labels. Benchmark: evaluation by regime, section reliability, and cross-tower transfer separate global accuracy from boundary reliability.

## 2 Related Work

**Benchmarks for scientific ML in engineering.** Recent scientific-ML benchmarks span PDE and multi-physics surrogates (PDEBench, PINNacle, The Well, LIPS) (Takamoto et al., 2022; Hao et al., 2024; Ohana et al., 2024; Leyli-Abadi et al., 2022), aerodynamics (AirfRANS, DrivAerNet++, CarBench) (Bonnet et al., 2022; Elrefaie et al., 2024; 2025), geophysics and weather (OpenFWI, WeatherBench 2) (Deng et al., 2022; Rasp et al., 2024), catalyst chemistry (OC20) (Chanussot et al., 2021), and inorganic materials property prediction (Matbench) (Dunn et al., 2020). These benchmarks have established the practice of releasing simulation and physical-data benchmarks at scale with reproducible splits and metrics. None, however, covers wind-turbine fatigue, and none partitions test points by their position relative to the training envelope (AirfRANS holds out Reynolds and angle-of-attack bands).

**Surrogate evaluation by operating-envelope coverage.** Engineering surrogates are deployed over physical operating envelopes (wind/wave states, flight conditions, soil parameters), and the practical question is whether predictions stay reliable as inputs drift away from the training cloud. Alpha shapes (Edelsbrunner et al., 1983) estimate this cloud's boundary from a finite sample, supporting a clean In-train / Interpolation / Extrapolation partition over the operating dimensions. Adopting this partition at the benchmark level shifts the leaderboard from average accuracy to boundary reliability, where deployment risk concentrates. Distribution-shift benchmarks, tabular (TableShift, WhyShift) and multi-domain (WILDS, Shifts) (Gardner et al., 2023; Liu et al., 2023; Koh et al., 2021; Malinin et al., 2021) split in- vs. out-of-distribution by metadata, time, or domain. Tabular leaderboards (TabArena, TALENT) (Erickson et al., 2025; Liu et al., 2025) use i.i.d. splits. FLOATBench instead places each test point by its position relative to the training domain (in-train, interpolation, extrapolation), so extrapolation depth is controlled by construction. It also adds a controlled shift across tower geometries. Tree ensembles are piecewise constant beyond the training range, since each leaf predicts a constant, while ReLU networks converge to linear functions outside the training support (Xu et al., 2021). Model selection under shift is itself hard: without test-distribution data, in-distribution validation can trail oracle selection (Gulrajani & Lopez-Paz, 2021). FLOATBench measures what these effects cost on a physical operating envelope: which surrogate validation selects, and how far it falls behind at the envelope edge.

**Datasets and surrogates for wind/offshore fatigue.** Wind-turbine fatigue datasets are the closest domain-specific resources, but public benchmark-ready releases remain limited (Table 1): prior releases ship one to three geometries of 5 to 10 MW, $\leq$ 2 tower sections, and a random or site-specific evaluation. The closest public analogue is the release of Papi & Bianchini (2024): $\sim$447 k hindcast-driven (historical site records) damage equivalent load (DEL) records for a single NREL 5 MW + OC4-DeepCwind semisubmersible configuration; no public release exists at the 20+ MW scale where current floating designs are converging. On the methods side, prior surrogate studies for wind/offshore fatigue span Kriging, Gaussian processes, polynomial chaos, neural networks, mixture-density networks, and copula-based models (Dimitrov et al., 2018; Slot et al., 2020; Haghi & Crawford, 2024; Singh et al., 2024; Liu et al., 2024; Singh et al., 2025; Ren & Xing, 2023; Müller & Cheng, 2018; Yuan et al., 2024; Zhao et al., 2024; Schmidt et al., 2025; Avendaño-Valencia et al., 2021), but each study uses its own simulation set, split, metrics, and reporting protocol, so the relative merits of these families remain unaudited and the field cannot adjudicate between them. The missing artifact is a shared benchmark that decouples the modeling question from the simulation question.

Table 1: Wind-turbine fatigue datasets: scope, envelope sampling, label coverage, and accessibility.

| Reference | Power [MW] | Substructure | #geom | #cond (input space) | #sect | #labels | Regime | Public |
|---|---|---|---|---|---|---|---|---|
| *Onshore (aero only)* | | | | | | | | |
| Dimitrov et al. (2018) | 10 | Onshore | 1 | 10 k (9-D) | 2 | 160 k | ✗ | ✗ |
| Slot et al. (2020) | 5 | Onshore | 1 | 0.6 k (5-D) | 2 | 125 k | ✗ | ✗ |
| IWT 7.5 MW (Requate & Meyer, 2023) | 7.5 | Onshore | 1 | 15 k (4-D) | 1 | 47 k | ✗ | ✓ |
| Haghi & Crawford (2024) | 5 | Onshore | 1 | 33 k (3-D) | 2 | 65 k | ✗ | ✓ |
| *Bottom-fixed offshore (aero + hydro on a static substructure)* | | | | | | | | |
| Müller & Cheng (2016) | 5 | Jacket | 1 | 0.7 k (5-D) | 2 | 9 k | ✗ | ✗ |
| Singh et al. (2024) | 10 | Onshore + monopile | 2 | 9 k (3-D/6-D) | 2 | 107 k | ✗ | ✓ |
| *Floating offshore (aero + hydro + mooring restoring and rigid-body floater motions)* | | | | | | | | |
| FLOATECH D2.3 (Papi et al., 2023) | 5/10 | Semisub. + Spar + Hexa | 3 | 0.5 k (5-D) | 2 | 3 k | ✗ | ✓ |
| Liu et al. (2024) | 5 | Semisub. (OC4 DeepCw.) | 1 | 0.2 k (3-D) | 1 | 0.2 k | ✗ | ✗ |
| Papi & Bianchini (2024) | 5 | Semisub. (OC4 DeepCw.) | 1 | 447 k (hindcast) | 2 | 894 k | ✗ | ✓ |
| Singh et al. (2025) | 6 | Spar (Hywind Scotland) | 1 | 9 k (7-D) | 2 | 22 k | ✗ | ✗ |
| **FLOATBench (ours)** | **22** | **Semisub. (IEA-22)** | **3** | **1,078 (3-D)** | **30** | **582 k** | ✓ | ✓ |

*#geom*: distinct turbine/substructure geometries; *#cond*: distinct operating conditions; *#sect*: tower cross-sections with fatigue labels; *#labels*: per-simulation, per-section fatigue labels.

FLOATBench provides, to our knowledge, the first public 22 MW FOWT fatigue benchmark with section-resolved labels: three 22 MW FOWT geometries (vs. one to three 5 to 10 MW geometries in prior releases), 30 tower sections (vs. 1–2), 582,120 per-section fatigue labels, and a regime-aware evaluation protocol derived from physical-envelope coverage rather than a random split. No public release combines a designed envelope grid (regimes defined by construction), matched geometries of one family, and section-resolved labels. The largest floating one (Papi & Bianchini, 2024) is hindcast-driven and follows one site's history. FLOATBench samples the joint wind–wave operating distribution on a designed grid, which controlled extrapolation tests of ML models require.

## 3 DATASET

**Operating envelope.** Each tower is simulated across a $22 \times 7 \times 7$ grid of environmental conditions (Figure 2): 22 mean hub-height wind speeds at the midpoints of 1 m/s bins from cut-in (3 m/s) to cut-out (25 m/s), and, for each wind speed, seven $H_s$ levels with seven $T_p$ levels per $H_s$, sampled from the joint wind–wave operating distribution used by Alves Ribeiro et al. (2026), yielding 49 sea states per wind speed and 1,078 unique (wind, wave) operating points per aligned.

**Simulation setup.** All simulations follow IEC 61400-3-2 (IEC, 2019) design load case (DLC) 1.2 (normal power production), run with OpenFAST (Jonkman, 2013) across the operating envelope. Each operating point is simulated with 6 independent turbulence (wind) seeds, giving 6,468 ten-minute simulations per tower ($1{,}078 \times 6$); runs last 1,000 s with the final 600 s post-processed. Each run records 88 time-series at 10 Hz organized into three groups (Figure 3): general turbine outputs (e.g., rotor speed, blade pitch, generator power, rotor thrust), tower outputs (e.g., section bending moments, accelerations, and deflections at multiple heights), and platform outputs (e.g., 6-DOF translation and rotation motions, mooring fairlead and anchor tensions, wave elevation), totaling $\approx$190 GB across the 19,404 runs for the 3 towers.

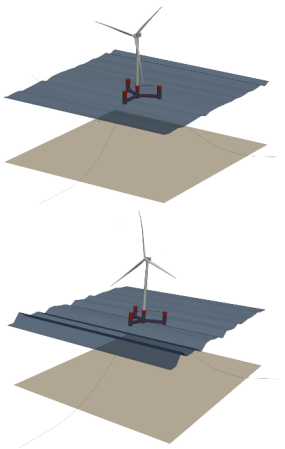


Figure 2: OpenFAST simulations near the envelope extremes: $V \approx 4.5$ m/s (top) and 24.5 m/s (bottom).

**Compute.** Simulations were executed on a commercial cloud HPC platform on 2-vCPU virtual-machine instances at $\approx$30 min wall-clock per run ($\approx 1$ CPU-hour). Each tower required 6,468 simulations ($\approx$ 6,500 CPU-hours, $\approx 0.74$ core-years), and the full 3-tower release consumed $\approx$19,400 CPU-hours in total ($\approx$2.2 core-years).

**Damage outputs.** Fatigue is quantified by the cumulative damage $D$. For each tower, each operating condition, and each of the 30 tower sections, OpenFAST produces a bending moment time-series over the post-processed window, which is converted to stress and processed through rainflow counting (Matsuishi & Endo, 1968; Downing & Socie, 1982) and Miner summation against the DNV-RP-C203 S–N curve E (DNV, 2024), equivalent to Eurocode 3 detail category 80 (bilinear with $m = 3/5$, knee at $10^7$ cycles, 25 mm thickness-correction reference), yielding the cumulative damage $D$, the released label. Pipeline overview in Figure 3; Appendix M reproduces the labels from the raw outputs. The load-equivalent transform $\mathrm{DEL} \propto D^{1/m}$, $m = 3$, computed from $D$, is also used as a regression target.

**Tower geometries.** Three towers are released, all on the same three-column semi-submersible platform, controller, and mooring: REF, the IEA-22-280-RWT (Zahle et al., 2024) baseline tower (designed for fixed-bottom conditions without explicit fatigue constraints (Alves Ribeiro et al., 2026), $D \approx 32$ at the tower base, $\approx 9$ months under Miner's rule on a 25-yr horizon); OPT1, an intermediate fatigue-aware re-design produced by the FLOAT method (Alves Ribeiro et al., 2026) ($D$ from 0.49 to 1.28 along the height); and OPT2, a final iterate targeting $D \leq 0.90$ (released labels 0.78 to 0.94). The redesign sequence REF $\rightarrow$ OPT1 $\rightarrow$ OPT2 varies the outer diameter and wall thickness profiles at constant tower height (149.386 m) (Figure 4). All three towers are discretized into 30 sections with damage labels at midpoints. Geometric profiles, damage profiles, and natural frequencies are in Appendix E.1.

**Dataset schema and access.** Each tower contributes 194,040 rows ($6{,}468 \times 30$ sections), totaling 582,120 rows across the three towers; we release three CSVs per tower: the training set, the test set with regime labels, and a raw table holding all rows without split or regime labels (the canonical source for reproducing the partition). Each row carries identifiers (simulation, section, and grid coordinates on the $22 \times 7 \times 7$ wind/wave envelope plus the turbulence-seed index), environmental features (nominal, mean, and std hub-height wind speed, $H_s$, $T_p$), tower section geometry (height, radius, thickness), per-row regime labels on test rows (wind and wave), the Miner-summed damage label, and a per-condition lifetime weighting factor for aggregating damage over the 25-year service life; full schema in Appendix E.2.

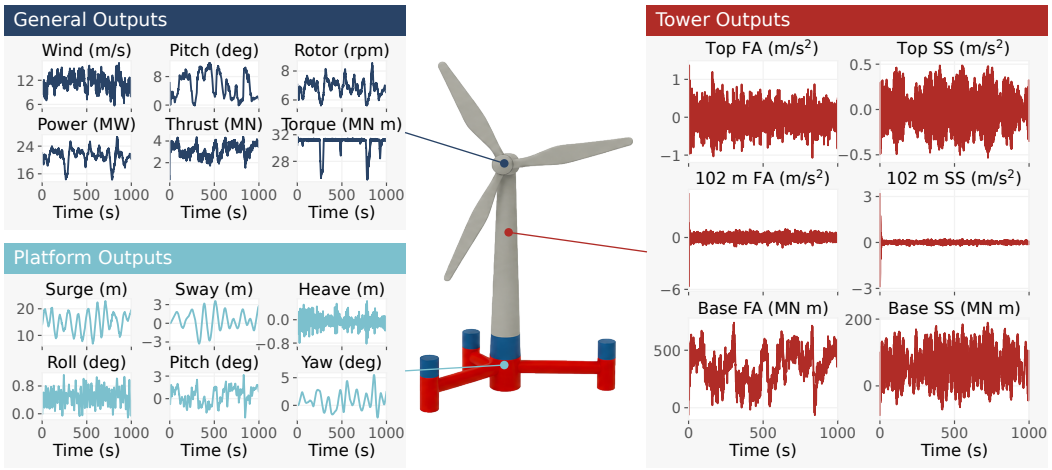
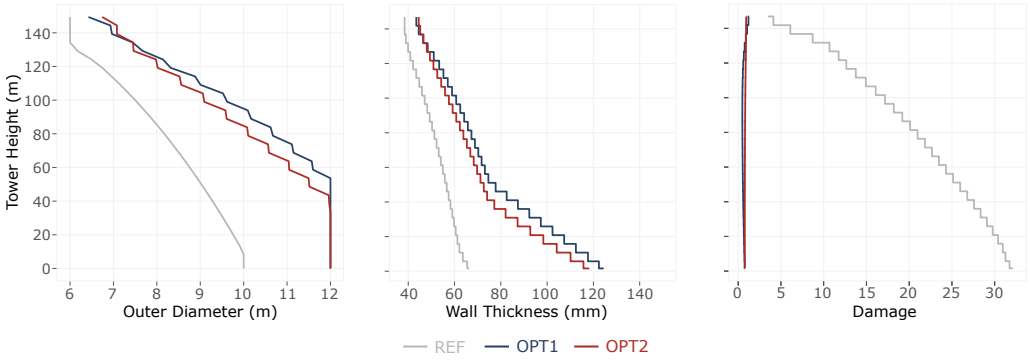



Figure 3: **FLOATBench OpenFAST outputs and damage pipeline.** Each of our OpenFAST simulations yields 88 time-series at 10 Hz for the IEA-22-280-RWT on a three-column semi-submersible. *Pipeline:* our OpenFAST simulations → 88 time-series → bending moments → stress → rainflow → S-N + Miner → damage labels.




(a) Outer diameter profile (b) Wall thickness profile (c) Damage profile

Figure 4: Geometry and FLOATBench lifetime damage profiles along the tower height for the three released towers REF, OPT1, and OPT2.

**Hosting and licensing.** The dataset is hosted on a public data repository (anonymized for review) under CC-BY-4.0; the release includes a Croissant 1.1 manifest and a Datasheet for Datasets (Gebru et al., 2021) (Appendix N). The benchmark code is released under MIT license. See Appendices A–C for details.

## 4 Benchmark Protocol

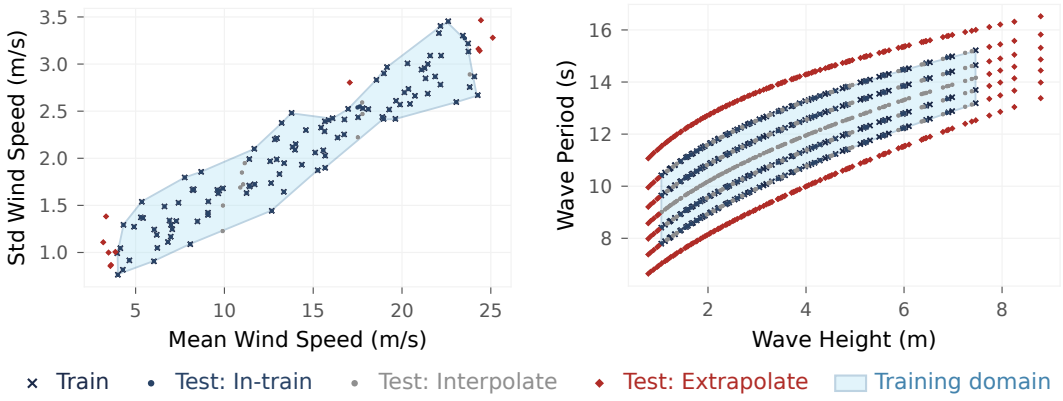
### 4.1 Regime-aware partition

**Train/test split.** From the $22 \times 7 \times 7$ grid, test holds the extreme and middle indices on both the wind and wave axes, leaving 18 of 22 wind speeds and a $4 \times 4$ inner wave patch on the train side (all 6 seeds per condition kept in the same fold to prevent train-test leakage; full index list in Appendix F.1). This yields 51,840 train / 142,200 test rows per tower ($\approx 27/73$ train/test split). Unlike a row-matched i.i.d. 80/20 baseline, the partition induces a structured covariate shift along both wind and wave axes; the test set is larger than the training set and lies mostly outside its support, keeping every extrapolation region densely sampled.

**Regime labeling.** Each test point is labeled In-train, Interpolation, or Extrapolation (IT, IP, EX) independently in the wind $(\text{mean}\, V, \text{std}\, V)$ and wave $(H_s, T_p)$ subspaces (Figure 5):

1. **Train spacing (train→train).** For each training point, the nearest-neighbor distance to the rest of the training set is computed in standardized feature units; the mean over all training points sets the spacing scale.
2. **Distance-based grouping (test→train).** For each test point, the nearest-neighbor distance to the training set is computed in standardized feature units and normalized by the spacing scale; points with normalized distance $\leq 0.5$ are labeled IT, the rest provisionally IP.
3. **Boundary-based override.** A concave hull ($\alpha$-shape, $\alpha = 0.1$) fitted in original (unstandardized) units bounds the training support in the wind and wave subspaces; test points outside the hull *and* at least $\varepsilon = \tau s$ from its boundary ($\tau = 0.5$ is the IT threshold, $s$ the standardized spacing scale), with the boundary distance measured in original units ($\varepsilon = 0.049$ in the wind plane, $0.021$ in the wave plane) are re-labeled EX. Points just outside the hull but within the tolerance stay as IP, so the tolerance acts as a slack band against boundary-proximate misclassification.

Pseudocode and method details are in Appendix F.2. Crossing the per-axis labels (each IT / IP / EX) gives nine joint regimes, with EX_EX (extrapolation on both axes) the deepest cell. Random validation samples test points from the training envelope and leaves every extrapolation regime empty by construction; the regime-aware partition populates all nine (random vs regime-aware composition in Appendix F.3).




(a) Wind subspace (b) Wave subspace

Figure 5: FLOATBench regime partition; the training domain (alpha-shape hull) is shaded.

### 4.2 Evaluation levels

**E1: baseline (random split).** A random partition matched in size to E2 ($\approx 27/73$ train/test split), run independently on each tower (REF, OPT1, OPT2). The same surrogate set and training-set size are used in both E1 and E2, so any performance difference between them reflects the partitioning strategy alone.

**E2: within-tower (by regime and section).** Models are evaluated under the regime-aware partition, run independently on each tower (REF, OPT1, OPT2); results are reported per regime (with EX_EX as the deepest extrapolation) and per section (Section 1 base vs Section 30 top).

**E3: cross-tower transfer.** Within-family cross-design transfer: a leave-one-tower-out scheme trains on two towers and evaluates on the held-out third (REF+OPT1→OPT2, REF+OPT2→OPT1, OPT1+OPT2→REF), measuring whether performance survives a change of geometry while holding the operating envelope fixed.

### 4.3 TABULAR SURROGATES

We evaluate the same surrogate models across all three levels (E1, E2, E3): XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), CatBoost (Prokhorenkova et al., 2018), NeuralNetFastAI (Howard & Gugger, 2020), NeuralNetTorch (Erickson et al., 2020), ExtraTrees (Geurts et al., 2006), RandomForest (Breiman, 2001), and TabM (Gorishniy et al., 2025). To cover a broad set of surrogates at each (tower-configuration, experiment level) combination, all baselines are trained through AutoGluon-Tabular 1.5.0 (Erickson et al., 2020), which provides pre-tuned hyperparameter portfolios and automatic stacking. We use the `best` (`hyperparameters=zeroshot`) and `extreme` (`hyperparameters=zeroshot_2025_tabfm`) presets, taking as inputs four environmental conditions (mean wind speed, std wind speed, $H_s$, $T_p$) and three section-geometry descriptors (section height, section radius, section thickness; full schema in Appendix E.2), and the cube-root damage transform $D^{1/3}$ (i.e. the DEL-format target) as the regression target. Each (tower-configuration, experiment level, preset) combination runs under a 4-hour wall-clock budget, yielding 82–96 models per tower for E1/E2 and 58–63 per fold for E3. Inside AutoGluon, TabPFN-v2 and Mitra abort at a 10k-row cap and TabICL is classification-only (Appendix F.4). TabPFN-2.5 is evaluated as a standalone baseline on the full training set (Appendix I); other foundation models remain future work. More on baselines and training in Appendix F.4 and Appendix F.5.

### 4.4 METRICS AND STATISTICAL SIGNIFICANCE

We assess surrogate predictions of damage (in DEL format) using two error metrics: **Rel L$^2$**, the relative $L^2$-norm error, captures global prediction quality on a dimensionless scale and is used as the primary metric for ranking models; **MRE** (point-wise mean relative error) captures the average per-point degradation and is reported for per-regime diagnostics. Lower is better in both. Auxiliary error metrics (MSE, MAE, RMSE, $R^2$, MaxErr) and compute metrics (inference latency, throughput, training time) are released in the leaderboard CSVs. Every leaderboard number is reported as $\bar{\theta}_{\text{boot}} \pm \sigma_{\text{boot}}$ (bootstrap mean and standard deviation, $B = 2{,}000$ resamples of whole operating conditions); $95\%$ percentile intervals are in the leaderboard CSVs and Appendix G.4. Formulas and the bootstrap algorithm are in Appendix F.6 and Appendix F.7.

## 5 RESULTS

Results follow the three protocol levels (Section 4.2). Full top-10 leaderboards for E1, E2, and E3 are in Appendix G, reported as bootstrap mean $\pm$ standard deviation; $95\,\%$ percentile intervals for the top-3 are in Appendix G.4.

### 5.1 RANDOM SPLIT (E1): EVERYTHING LOOKS EASY, THE WRONG MODELS WIN

Under E1's random split, all top-10 surrogates (Appendix G.1) appear strong: roughly 2–4$\%$ Rel L$^2$ DEL across towers, with bootstrap CIs overlapping between adjacent ranks. But the test set contains no hard regime: random validation leaves every extrapolation regime empty (see Appendix F.3 for regime composition), so the regimes that matter are invisible by construction. A condition-grouped random split is equally optimistic ($0.0198$ vs. $0.0191$ on REF; Appendix H.4): the cause is absent extrapolation, not seed leakage. The leaderboard rewards fit inside the training domain, and the rank-1 surrogate under E1 is *not* the rank-1 at EX_EX.

*A damage benchmark needs a regime-aware split, not a random one, to choose the right model.*

### 5.2 WITHIN-TOWER REGIME-AWARE (E2): DIFFERENT REGIMES PICK DIFFERENT WINNERS

**The Global winner loses at the wind-and-wave extrapolation regime.** EX_EX (extrapolation on both wind and wave) is the deepest extrapolation cell and has the highest MRE on every tower (Appendix K.1). Ranking on all test points (Global) versus on EX_EX points only gives different winners on every tower: the Global rank-1 `WeightedEnsemble_L2` drops to EX_EX ranks 23 / 11 / 11 (REF / OPT1 / OPT2; on REF it is statistically tied with rank 2 globally, Table 15), while the EX_EX rank-1 `NeuralNetFastAI_r102_BAG_L1` sits at Global ranks 79 / 73 / 69 (Figure 6). We call this a rank reversal.

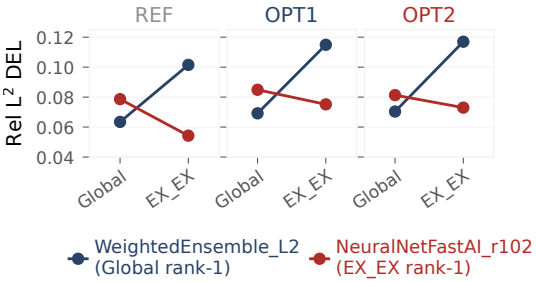



Figure 6: **Within-tower rank reversal (E2):** Rel $L^2$ DEL, Global vs. EX_EX. The Global winner drops at EX_EX, where a low-ranked network wins.

Table 2: **Main results (E2):** Rel $L^2$ DEL, Global / EX_EX. †Global and *EX_EX rank-1 of the AutoGluon pool; other rows: five of the seven standalone baselines; all nine regimes in Appendix I.

| Model | REF | OPT1 | OPT2 |
|---|---|---|---|
| Ensemble† | **0.063** / 0.101 | **0.069** / 0.115 | **0.070** / 0.117 |
| FastAI NN* | 0.079 / **0.054** | 0.085 / **0.075** | 0.081 / **0.073** |
| TabPFN | 0.072 / 0.104 | 0.072 / 0.099 | 0.071 / 0.090 |
| XGBoost | 0.078 / 0.115 | 0.078 / 0.150 | 0.079 / 0.153 |
| PCE | 0.087 / 0.074 | 0.108 / 0.117 | 0.104 / 0.118 |
| Resp. surface | 0.087 / 0.073 | 0.143 / 0.194 | 0.137 / 0.189 |
| GP | 0.131 / 0.307 | 0.147 / 0.340 | 0.145 / 0.335 |

**The rank reversal is robust and not an AutoML artifact.** It holds for every tested setting of the partition parameters (alpha, distance threshold, tolerance) that still leaves an extrapolation region, on all three towers, and after retraining the full pool on two other training grids (REF); it grows stronger the deeper the extrapolation (Appendix H). It also holds outside AutoML (Table 2). TabPFN and tuned XGBoost do not beat the EX_EX winner. Polynomial surrogates are weak globally and the GP degrades sharply at EX_EX. Selecting on validation, the standard option without extrapolation labels, leaves 1.5–1.9× the best available EX_EX error. The ensemble puts 86–88% of its weight on tree models, which under-predict beyond the training domain (Appendix J).

**The rank reversal is a family effect; TabM degrades most.** A tree-weighted ensemble wins globally and bagged networks, a high-variance family, win at EX_EX. The EX_EX top-10 (Appendix K.2) is dominated by `NeuralNet` BAG_L1 variants (FastAI and Torch backends) across all three towers. The per-model scatter of MRE DEL on Global vs. EX_EX (Appendix K.3) confirms the family pattern: `NeuralNet` variants stay closest to the diagonal (similar errors on Global and EX_EX), while `TabM` falls farthest below (heaviest degradation at EX_EX).

**Wind drives the extrapolation; wave axis stays flat.** Aggregating errors by model family (see Appendix K.4 for family-aggregated MRE DEL by wind and wave regime), the wind axis inflates by 6–20× from In-train to Extrapolation, while the wave axis stays flat (1.1–1.4×). Under Rel $L^2$ too, wave extrapolation alone leaves the ensemble at In-train error (0.047 vs. 0.048 on REF; Table 22). `NeuralNet` is the best family on wind Extrapolation and `TabM` the worst.

**Sections pick different winners than Global.** Ranking on each section's test points alone gives different winners than the Global ranking. On REF, no model wins both sections: `NeuralNetFastAI_r11_BAG_L2` wins Section 1 (Global rank 5), `RandomForestMSE_BAG_L1` wins Section 30 (Global rank 60). On OPT1/OPT2, `WeightedEnsemble_L2` (the Global rank-1) keeps rank-1 at the base (Section 1) but loses Section 30 to `NeuralNetTorch_r22_BAG_L1` (Global ranks 71/44). Full per-section top-10 in Appendix K.5.

*For damage prediction, the Global rank-1 does not remain best across regimes and sections: it loses at* EX_EX *and at the tower top, while remaining strongest at the base for* OPT1/OPT2.

### 5.3 Cross-tower transfer (E3): the rank-1 fails again, on geometry and on sections

**Transfer collapses on geometries far from training.** E3 trains on two towers and evaluates on the held-out third (E3 top-10 per fold in Appendix G.3). The cross-tower error depends sharply on which towers are in training (Figure 7): training on a set that includes REF generalizes to the re-designed geometries with rank-1 Rel $L^2$ DEL of 0.067/0.098 (REF+OPT1 → OPT2 and REF+OPT2 → OPT1), while training without REF (OPT1+OPT2 → REF) reaches 0.423, a 4–6× increase. Holding REF out under-predicts, consistent with its wider damage profile and most-distinct geometry (Figure 4, Appendix L.1).

**Sections pick different winners than Global, across folds.** On every fold, no model wins all three rankings: the Section 1 (base) winner is a `NeuralNet` variant on all three folds, while `TabM_r52_BAG_L1` wins Section 30 (top) on all three. The Global winner is `TabM_r52_BAG_L1` on the two folds whose training set includes REF (statistically tied with `NeuralNetFastAI_r191` on REF+OPT2 → OPT1), and `NeuralNetFastAI_r191_BAG_L1` on OPT1+OPT2 → REF. Full per-section top-10 in Appendix L.2.

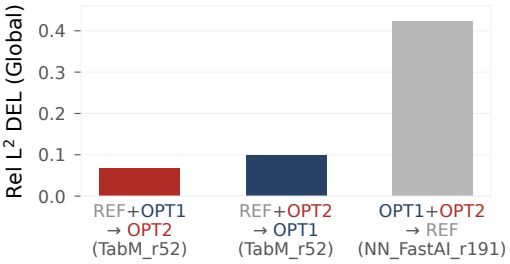



Figure 7: **Cross-tower transfer (E3):** rank-1 Rel $L^2$ DEL per fold (model name in parentheses).

*For damage prediction across towers, the Global rank-1 again does not remain best across geometry and sections.*

## 6 Scope, Design Choices, and Future Work

**Three geometries, one design family.** The three towers are the IEA-22 baseline and two fatigue-aware re-designs of it: the setting of design iteration, where a surrogate must follow the design as it changes. They are not near-duplicates: the first fore-aft frequency of REF lies close to the 3P (blade-passing) excitation frequency while the re-designs sit above it (Table 5), and the base damage changes by a factor of about 40 (Table 4). E3 therefore tests within-family transfer under a real dynamic shift, over three leave-one-tower-out folds. It does not establish cross-family generalization (e.g., to NREL 5 MW, DTU 10 MW, or IEA-15); that is left to future releases. The regime partition depends only on the wind/wave grid, so it carries over unchanged to other families.

**Load case and fatigue labels follow design practice.** Fatigue surrogates are built per design and must hold across the operating envelope, which is why the envelope, not the geometry count, is the primary out-of-distribution axis of the benchmark. The first release covers aligned wind-wave power production (DLC 1.2), the standard fatigue design load case of IEC 61400-3-2 (IEC, 2019), which typically dominates lifetime tower fatigue (Alves Ribeiro et al., 2026; Pacheco et al., 2023). The S–N curve is fixed by DNV-RP-C203 (DNV, 2024) for the welded tower detail; other curves change only the label-side step and can be applied through the audit subset (Appendix M). Misaligned, parked, and idling load cases involve different physics and new simulation campaigns; we draw no conclusions there.

**Future work.** Further tabular foundation models (TabICL, TabDPT); a time-series task on the ≈190 GB OpenFAST record (a 147-simulation audit subset is already released, Appendix M); other load cases and design families; and FEA/CFD meshes for multi-fidelity benchmarks.

## 7 Conclusion

**FLOATBench** releases 582,120 per-section fatigue-damage labels over three 22 MW floating-tower geometries, a regime-aware alpha-shape partition of the joint wind/wave envelope, and three protocol levels. Across 735 trained surrogates and seven standalone baselines, the regime-aware protocol shows that the global rank-1 surrogate is not the rank-1 at the deepest extrapolation cell. Random splits miss this rank reversal. It survives every tested partition setting that keeps an extrapolation region and two alternative training grids, and validation-based selection does not find the extrapolation winner. The regime-aware protocol applies to tabular surrogates trained on designed samples of a bounded operating envelope.

### Ethics Statement

FLOATBench contains simulated engineering data only: no human subjects, personal data, or proprietary measurements. Data are released under CC-BY-4.0 and code under the MIT license;

third-party assets and their licenses are listed in Appendix C. Surrogates trained on the benchmark are not a substitute for certification analyses; the regime-aware protocol is intended to expose where they are unreliable. Broader impact is discussed in Appendix D.

### Reproducibility statement

The dataset (per-section labels, regime labels, splits), the evaluation and training code, and the condition-level bootstrap script are provided through the anonymized links of Appendix B; the Croissant 1.1 manifest and the trained surrogates (5–10 GB) are released on publication and can be regenerated with the released code. Appendix A and Appendix B describe the artifacts and hardware; Appendix E the schema and splits; Appendix F.6 and Appendix F.7 the metrics and bootstrap; Appendix F.5 the training configuration; and Appendix M a raw time-series subset with a runner that reproduces the released labels from the simulator outputs.

## References

João Alves Ribeiro, Bruno Alves Ribeiro, Francisco Pimenta, Sérgio M. O. Tavares, Jie Zhang, and Faez Ahmed. Offshore wind turbine tower design and optimization: A review and AI-driven future directions. *Applied Energy*, 397:126294, 2025. doi: 10.1016/j.apenergy.2025.126294.

João Alves Ribeiro, Francisco Pimenta, Bruno Alves Ribeiro, Sérgio M. O. Tavares, and Faez Ahmed. FLOAT: Fatigue-aware design optimization of floating offshore wind turbine towers. *Ocean Engineering*, 362:126285, 2026. doi: 10.1016/j.oceaneng.2026.126285.

Luis David Avendaño-Valencia, Imad Abdallah, and Eleni Chatzi. Virtual fatigue diagnostics of wake-affected wind turbine via Gaussian process regression. *Renewable Energy*, 170:539–561, 2021. doi: 10.1016/j.renene.2021.02.003.

Florent Bonnet, Jocelyn Ahmed 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 (NeurIPS), Datasets and Benchmarks Track*, volume 35, pp. 23463–23478, 2022.

George E. P. Box and K. B. Wilson. On the experimental attainment of optimum conditions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(1):1–38, 1951. doi: 10.1111/j.2517-6161.1951.tb00067.x.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004. doi: 10.1145/1015330.1015432.

Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Rivière, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. Lawrence Zitnick, and Zachary Ulissi. Open Catalyst 2020 (OC20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, 2021. doi: 10.1021/acscatal.0c04525.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.

Chengyuan Deng, Shihang Feng, Hanchen Wang, Xitong Zhang, Peng Jin, Yinan Feng, Qili Zeng, Yinpeng Chen, and Youzuo Lin. OpenFWI: Large-scale multi-structural benchmark datasets for full waveform inversion. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 35, pp. 6007–6020, 2022.

Nikolay Dimitrov, Mark C. Kelly, Andrea Vignaroli, and Jacob Berg. From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases. *Wind Energy Science*, 3(2):767–790, 2018. doi: 10.5194/wes-3-767-2018.

DNV. DNV-RP-C203: Fatigue design of offshore steel structures. Recommended practice, DNV AS, Høvik, Norway, 2024.

S. D. Downing and D. F. Socie. Simple rainflow counting algorithms. *International Journal of Fatigue*, 4(1): 31–40, 1982. doi: 10.1016/0142-1123(82)90018-4.

Alexander Dunn, Qi Wang, Alex Ganose, Daniel Dopp, and Anubhav Jain. Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm. *npj Computational Materials*, 6(1):138, 2020. doi: 10.1038/s41524-020-00406-3.

Herbert Edelsbrunner, David G. Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983. doi: 10.1109/TIT.1983.1056714.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. doi: 10.1214/aos/1176344552.

Bradley Efron. Nonparametric standard errors and confidence intervals. *Canadian Journal of Statistics*, 9(2): 139–158, 1981. doi: 10.2307/3314608.

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 (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 499–536, 2024.

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

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. AutoGluon-Tabular: Robust and accurate AutoML for structured data. arXiv preprint arXiv:2003.06505, 2020.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 38, pp. 17285–17350, 2025. doi: 10.52202/085713-0519.

C. A. Field and A. H. Welsh. Bootstrapping clustered data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(3):369–390, 2007. doi: 10.1111/j.1467-9868.2007.00593.x.

Joshua P. Gardner, Zoran Popović, and Ludwig Schmidt. Benchmarking distribution shift in tabular data with TableShift. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 36, pp. 53385–53432, 2023. doi: 10.52202/075280-2324.

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.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006. doi: 10.1007/s10994-006-6226-1.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *International Conference on Learning Representations (ICLR)*, 2025.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the state of the art in tabular foundation models. arXiv preprint arXiv:2511.08667, 2025.

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

Rad Haghi and Curran Crawford. Data-driven surrogate model for wind turbine damage equivalent load. *Wind Energy Science*, 9(11):2039–2062, 2024. doi: 10.5194/wes-9-2039-2024.

Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, and Jun Zhu. PINNacle: A comprehensive benchmark of physics-informed neural networks for solving PDEs. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 76721–76774, 2024.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi: 10.1038/s41586-024-08328-6.

Jeremy Howard and Sylvain Gugger. fastai: A layered API for deep learning. *Information*, 11(2):108, 2020. doi: 10.3390/info11020108.

IEC. IEC TS 61400-3-2:2019. Wind energy generation systems – Part 3-2: Design requirements for floating offshore wind turbines. Technical specification, International Electrotechnical Commission, Geneva, Switzerland, 2019.

Jason M. Jonkman. The new modularization framework for the FAST wind turbine CAE tool. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2013. doi: 10.2514/6.2013-202. Software: OpenFAST v3.5, `https://github.com/OpenFAST/openfast`.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 3146–3154, 2017.

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 A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664, 2021.

Milad Leyli-Abadi, Antoine Marot, Jérôme Picault, David Danan, Mouadh Yagoubi, Benjamin Donnot, Seif-Eddine Attoui, Pavel Dimitrov, Asma Farjallah, and Clement Etienam. LIPS: Learning industrial physical simulation benchmark suite. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 35, pp. 28095–28109, 2022.

Ding Peng Liu, Giulio Ferri, Taemin Heo, Enzo Marino, and Lance Manuel. On long-term fatigue damage estimation for a floating offshore wind turbine using a surrogate model. *Renewable Energy*, 225:120238, 2024. doi: 10.1016/j.renene.2024.120238.

Jiashuo Liu, Tianyu Wang, Peng Cui, and Hongseok Namkoong. On the need for a language describing distribution shifts: Illustrations on tabular datasets. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, volume 36, pp. 51371–51408, 2023.

Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Huai-Hong Yin, Tao Zhou, Jun-Peng Jiang, and Han-Jia Ye. TALENT: A tabular analytics and learning toolbox. *Journal of Machine Learning Research*, 26(226):1–16, 2025.

Andrey Malinin, Neil Band, Yarin Gal, Mark Gales, Alexander Ganshin, German Chesnokov, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, Vyas Raina, Denis Roginskiy, Mariya Shmatova, Panagiotis Tigas, and Boris Yangel. Shifts: A dataset of real distributional shift across multiple large-scale tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

M. Matsuishi and T. Endo. Fatigue of metals subjected to varying stress. In *Proceedings of the Kyushu Branch of the Japan Society of Mechanical Engineers*, pp. 37–40, Fukuoka, Japan, 1968.

Kolja Müller and Po Wen Cheng. Validation of uncertainty in IEC damage calculations based on measurements from alpha ventus. *Energy Procedia*, 94:133–145, 2016. doi: 10.1016/j.egypro.2016.09.208.

Kolja Müller and Po Wen Cheng. A surrogate modeling approach for fatigue damage assessment of floating wind turbines. In *Proceedings of the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering (OMAE), Volume 10: Ocean Renewable Energy*, pp. V010T09A065, 2018. doi: 10.1115/OMAE2018-78219.

Ruben Ohana, Michael McCabe, Lucas Thibaut Meyer, Rudy Morel, Fruzsina Julia Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond Buschman Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich Kerswell, Suryanarayana Maddu, Jonah M. Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam Holden 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 (NeurIPS), Datasets and Benchmarks Track*, volume 37, pp. 44989–45037, 2024.

João Pacheco, Francisco Pimenta, Sérgio Pereira, Álvaro Cunha, and Filipe Magalhães. Experimental evaluation of strategies for wind turbine farm-wide fatigue damage estimation. *Engineering Structures*, 285:115913, 2023. doi: 10.1016/j.engstruct.2023.115913.

Francesco Papi and Alessandro Bianchini. 50 years of load simulation of the NREL 5MW OC4 floating wind turbine. Zenodo dataset, 2024. doi: 10.5281/zenodo.10514143.

Francesco Papi, Robert Behrens de Luna, Joseph Saverin, David Marten, Cyril Compbreau, Gerardo Mirra, Giancarlo Troise, and Alessandro Bianchini. FLOATECH Deliverable 2.3: Design load case database for code-to-code comparison. Zenodo dataset, 2023. doi: 10.5281/zenodo.8383686.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: Unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 6638–6648, 2018.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 50817–50847, 2025.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallègue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. WeatherBench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024. doi: 10.1029/2023MS004019.

Chao Ren and Yihan Xing. AK-MDAmax: Maximum fatigue damage assessment of wind turbine towers considering multi-location with an active learning approach. *Renewable Energy*, 215:118977, 2023. doi: 10.1016/j.renene.2023.118977.

Niklas Requate and Tobias Meyer. Database of short term damage equivalent loads (DEL) of IWT7.5MW wind turbine depending on wind, TI, yaw and derating. Zenodo dataset, 2023. doi: 10.5281/zenodo.8385296.

Franziska Schmidt, Clemens Hübler, and Raimund Rolfes. Kriging meta-models for damage equivalent load assessment of idling offshore wind turbines. *Wind Energy Science*, 10(12):3069–3089, 2025. doi: 10.5194/wes-10-3069-2025.

Deepali Singh, Richard P. Dwight, and Axelle Viré. Probabilistic surrogate modeling of damage equivalent loads on onshore and offshore wind turbines using mixture density networks. *Wind Energy Science*, 9(10): 1885–1904, 2024. doi: 10.5194/wes-9-1885-2024.

Deepali Singh, Erik Haugen, Kasper Laugesen, Richard P. Dwight, and Axelle Viré. Data-driven probabilistic surrogate model for floating wind turbine lifetime damage equivalent load prediction. *Wind Energy Science*, 10(12):2865–2888, 2025. doi: 10.5194/wes-10-2865-2025.

René M. M. Slot, John D. Sørensen, Bruno Sudret, Lasse Svenningsen, and Morten L. Thøgersen. Surrogate model uncertainty in wind turbine reliability assessment. *Renewable Energy*, 151:1150–1162, 2020. doi: 10.1016/j.renene.2019.11.101.

Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008. doi: 10.1016/j.ress.2007.04.002.

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 (NeurIPS), Datasets and Benchmarks Track*, volume 35, pp. 1596–1611, 2022.

Dongbin Xiu and George Em Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002. doi: 10.1137/S1064827501387826.

Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S. Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Xinyu Yuan, Qian Huang, Dongran Song, E Xia, Zhao Xiao, Jian Yang, Mi Dong, Renyong Wei, Evgeny Solomin, and Young-Hoon Joo. Fatigue load modeling of floating wind turbines based on vine copula theory and machine learning. *Journal of Marine Science and Engineering*, 12(8):1275, 2024. doi: 10.3390/jmse12081275.

Frederik Zahle, Thanasis Barlas, Kenneth Loenbaek, Pietro Bortolotti, Daniel Zalkind, Lu Wang, Casper Labuschagne, Latha Sethuraman, and Garrett Barter. Definition of the IEA Wind 22-Megawatt offshore reference wind turbine. Technical report, Technical University of Denmark, Kongens Lyngby, Denmark, 2024. doi: 10.11581/DTU.00000317. Code and data: `https://github.com/IEAWindSystems/IEA-22-280-RWT`.

Xiyuan Zhang, Danielle C. Maddix, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael W. Mahoney, Cuixiong Hu, Huzefa Rangwala, George Karypis, and Bernie Wang. Mitra: Mixed synthetic priors for enhancing tabular foundation models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 17831–17876, 2025. doi: 10.52202/085713-0535.

Guanhua Zhao, Sheng Dong, and Yuliang Zhao. Fatigue reliability analysis of floating offshore wind turbines under the random environmental conditions based on surrogate model. *Ocean Engineering*, 314:119686, 2024. doi: 10.1016/j.oceaneng.2024.119686.

## TABLE OF CONTENTS FOR APPENDICES

**A Reproducibility statement** ..... 16
**B URL and links** ..... 16
**C Author statement and data license confirmation** ..... 16
**D Broader impact** ..... 17
**E Dataset details** ..... 18
E.1 Tower geometry, damage, and natural frequencies (Figure 8, Table 3, Table 4, Table 5)
E.2 Schema reference (Table 6)
**F Benchmark details** ..... 21
F.1 Train/test selection (Table 7)
F.2 Regime-labeling algorithm and distance distributions (Algorithm 1, Figure 9, Table 8)
F.3 Random vs regime-aware split: regime composition (Table 9, Figure 10)
F.4 Baseline models (Table 10)
F.5 Training configuration
F.6 Metric definitions
F.7 Bootstrap confidence intervals (Algorithm 2)
**G Leaderboards (E1, E2 Global, E3 cross-tower)** ..... 28
G.1 Top-10 under random (E1), per tower (Table 12)
G.2 Top-10 under regime-aware (E2), per tower (Table 13)
G.3 Top-10 under cross-tower (E3), per fold (Table 14)
G.4 Bootstrap CIs for top-3 surrogates (Rel $L^2$, E2 and E3) (Table 16)
**H Is the rank reversal an artifact of the partition?** ..... 33
H.1 Regime labels are stable under the partition parameters
H.2 The rank reversal holds in every variant that keeps an extrapolation region
H.3 The rank reversal survives retraining on two other grids and grows with extrapolation depth
H.4 Random-split optimism comes from missing extrapolation, not seed leakage
**I Do models outside AutoML change the picture?** ..... 36
**J Can validation pick the extrapolation winner, and why do ensembles fail?** ..... 39
J.1 Validation selection leaves 1.5 to 1.9 times the best extrapolation error
J.2 The ensemble leans on trees, and trees under-predict beyond the training domain
**K Which surrogate wins depends on the regime and the section (E2)** ..... 41
K.1 Joint wind and wave extrapolation has the highest MRE on every tower (Figure 12)
K.2 Every EX_EX top-10 model is a bagged neural network (Table 25)
K.3 Neural networks degrade least from Global to EX_EX; TabM degrades most (Figure 13)
K.4 Wind extrapolation drives the error; the wave axis stays flat (Figure 14)
K.5 The base and the top pick different winners (Table 26, Table 27)
**L Does the winner survive a change of tower? (E3)** ..... 47
L.1 Transfer holds when REF is in training and under-predicts when it is held out (Figure 15)
L.2 TabM wins the top on every fold; neural networks win the base (Table 28, Table 29)
**M Can the labels be reproduced from the raw signals?** ..... 50
**N Datasheet for Datasets** ..... 51
**O AI use statement** ..... 56

## A REPRODUCIBILITY STATEMENT

We provide an anonymized code repository to facilitate the reproduction of all experiments in the main paper, along with links to download the dataset. The repository includes code to replicate the benchmark experiments and to generate the figures of the paper. The full dataset is available for review at the links provided in Appendix B. The training and experiments were performed on a machine equipped with an Intel Core i9-14900K processor (24 cores, 32 threads), 128 GB RAM, and an NVIDIA GeForce RTX 4090 GPU (24 GB VRAM).

## B URL AND LINKS

**Project page.** An anonymized project page gathers the links below, an interactive dataset explorer, the leaderboards and instructions to download the code and the data and run the benchmark, `https://anonymous.4open.science/w/FLOATBench-115C/review/index.html`.

**FLOATBench dataset.** The FLOATBench dataset will be publicly released under CC-BY-4.0 on publication. For review, the complete dataset (the per-tower CSVs and their README, folder `dataset/`) is available at the anonymized link `https://osf.io/te9na/?view_only=224887b50912448b871620b3ef96cefc`.

**Tower geometry artifacts.** The baseline IEA-22-280-RWT reference turbine is available at `https://github.com/IEAWindSystems/IEA-22-280-RWT`. The FLOAT-derived redesigns OPT1 and OPT2 are released on publication; for review, their section geometries are in the audit subset (OSF link above, folder `audit_package/geometry`).

**Croissant metadata.** A Croissant 1.1 metadata record documenting the dataset is released with the dataset on publication.

**Benchmark code.** The code for training, evaluation and figures is available at the anonymized repository `https://anonymous.4open.science/r/FLOATBench-84A1/`. A script in the code (`scripts/download`) downloads the dataset from the link above and checks that the file is intact.

**Raw time-series audit subset.** The raw OpenFAST signals of 147 simulations, their labels and the tower geometries (Appendix M) are available at `https://osf.io/te9na/?view_only=224887b50912448b871620b3ef96cefc` (folder `audit_package`); the post-processing code and the audit runner are in the benchmark code (`floatbench/openfast`, `scripts/analyses/audit`).

## C AUTHOR STATEMENT AND DATA LICENSE CONFIRMATION

We, the authors, hereby declare that we bear full responsibility for any violations of rights or other issues arising from the use of the data. We confirm that the FLOATBench dataset is released under the Creative Commons Attribution 4.0 International license (CC-BY-4.0), which allows others to share and adapt the work for any purpose, including commercial use, provided that appropriate credit is given. The benchmark code is released under the MIT license.

**Third-party assets and their licenses.** The existing assets used in this work are released under permissive open-source licenses, all compatible with the redistribution terms of FLOATBench: IEA-22-280-RWT (Zahle et al., 2024) (Apache-2.0); OpenFAST (Jonkman, 2013) (Apache-2.0); AutoGluon-Tabular 1.5.0 (Apache-2.0); XGBoost (Apache-2.0); LightGBM (MIT); CatBoost (Apache-2.0); scikit-learn (BSD-3-Clause), which provides the ExtraTrees and RandomForest baselines; the FastAI- and PyTorch-based neural-network learners included in AutoGluon's portfolio (Apache-2.0); and TabM (MIT). All licenses have been reviewed and respected.

## D BROADER IMPACT

Tower fatigue is the binding design constraint for the 22 MW floating turbines that will harvest the world's deep-water wind resource, and surrogate models are a practical way to make the design loop tractable. FLOATBench is, to the authors' knowledge, the first principled, multi-geometry, per-section fatigue benchmark for this problem, and its impact spans several research communities:

1. **Advancing floating wind design and accelerating design cycles:** Enabling detailed studies in tower fatigue to foster innovation in larger-class FOWT design, assisting structural engineers in creating more accurate surrogate models, and supporting the deployment of next-generation floating wind farms.
2. **Accelerating fatigue assessment:** FLOATBench provides damage labels computed from high-fidelity OpenFAST simulations, supporting the training of surrogate models that reduce the computational cost and time required for fatigue analysis and enable faster iteration over the design space.
3. **Machine learning integration:** Offering a rich resource for training and testing tabular surrogates on a physically structured regression task with realistic boundary regimes.
4. **Benchmarking and validation:** Serving as a benchmark dataset with a regime-aware evaluation protocol, aiding the validation of boundary reliability for surrogate models and exposing global-versus-boundary failures invisible under random splits; this matters operationally because real-world wind and wave conditions routinely fall at the boundary of a finite sampled training envelope, exactly where these failures become consequential.
5. **Environmental impact:** Contributing to the development of more cost-effective floating wind farms, supporting the deployment of deep-water renewable energy and reductions in levelized cost of energy.
6. **Safety and certification support:** Accelerating early design exploration with fast surrogate predictions while certification-grade DLC analysis remains required for final verification. The released labels use one standard fatigue configuration under DLC 1.2 only, so they are not a substitute for certification-grade analysis.
7. **Cross-disciplinary insights:** Offering insights at the intersection of structural fatigue, machine learning, and floating wind design, encouraging cross-disciplinary research and collaboration.
8. **Foundation for follow-up ML research:** The $\sim$190 GB OpenFAST time-series record (88 outputs at 10 Hz across the $19{,}404$ runs) used to compute the released tabular labels, together with the planned multi-fidelity FEA/CFD extensions (Section 6), positions FLOATBench as a foundation for follow-up ML research on time-series surrogates, neural operators, and multi-fidelity learning beyond the tabular task released here.

Beyond floating wind, FLOATBench provides a template for engineering surrogate tasks defined over designed samples of a bounded operating envelope: a regime-aware partition that exposes failures random splits cannot see, per-section resolution that connects model accuracy to component-level reliability, and multi-geometry evaluation that tests transfer across matched geometries within a design family. We expect this template to inform surrogate development in adjacent engineering domains where realistic deployment shifts make boundary reliability the operationally binding metric.

# E DATASET DETAILS

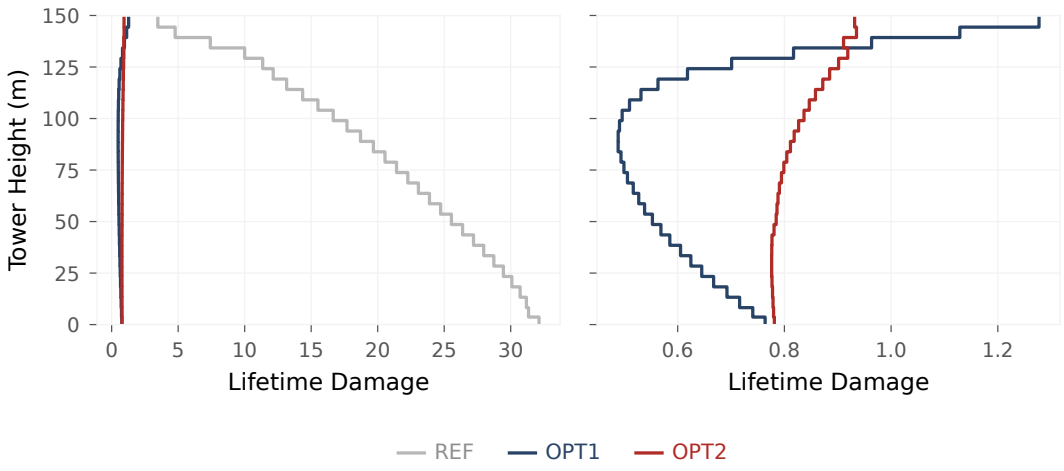
## E.1 TOWER GEOMETRY, DAMAGE, AND NATURAL FREQUENCIES

**Tower geometry.** Table 3 reports the diameter and wall thickness at the bottom and top of each released tower. Both optimized towers thicken toward the base and taper toward the top; OPT2 distributes material slightly more evenly than OPT1 (thinner base, thicker top). Full geometry artifacts (CAD profiles and OpenFAST input decks) for OPT1/OPT2 are released on publication (section geometries for review in the audit subset, Appendix B); upstream REF artifacts at `https://github.com/IEAWindSystems/IEA-22-280-RWT`.

Table 3: FLOATBench tower diameter and wall thickness at base and top for the three released towers; percentage change relative to REF in parentheses.

| Parameter | REF | OPT1 | OPT2 |
|---|---|---|---|
| Diameter, bottom (m) | 10.000 | 12.000 (+20.0 %) | 12.000 (+20.0 %) |
| Diameter, top (m) | 6.000 | 6.424 (+7.1 %) | 6.741 (+12.4 %) |
| Thickness, bottom (mm) | 66 | 124 (+87.6 %) | 118 (+78.2 %) |
| Thickness, top (mm) | 38 | 43 (+13.1 %) | 45 (+15.8 %) |
| Total mass (t) | 1,574 | 2,899 (+84.2 %) | 2,656 (+68.8 %) |

**Damage profile.** Figure 8 and Table 4 report the per-section lifetime weighted damage profile and the boundary values. REF varies by an order of magnitude along the height ($\approx 4$ to $\approx 32$); OPT1 varies modestly (0.488–1.277); OPT2 is nearly uniform (0.776–0.935), with 5 of 30 sections slightly above the 0.90 target (maximum 0.935 at 141.8 m). This progression reflects the FLOAT (Alves Ribeiro et al., 2026) re-design, which minimizes tower mass subject to a section-wise fatigue constraint of $D \leq 0.90$.




(a) All three towers

(b) Zoom on the two re-designs

Figure 8: FLOATBench lifetime weighted section damage along the tower.

Table 4: FLOATBench lifetime weighted section damage at base and top; percentage change relative to REF in parentheses.

| | REF | OPT1 | OPT2 |
|---|---|---|---|
| Damage, bottom | 32.128 | 0.764 (−97.6 %) | 0.781 (−97.6 %) |
| Damage, top | 3.471 | 1.277 (−63.2 %) | 0.932 (−73.1 %) |

**Natural frequencies.** Table 5 lists the first fore-aft (FA1) natural frequency per tower in the floating configuration. REF FA1 lies close to the 3P excitation frequency ($\approx 0.35$ Hz), indicating a potential

resonance risk under specific operating conditions; the FLOAT (Alves Ribeiro et al., 2026) re-design shifts FA1 above this frequency, into the stiff–stiff range between 3P and 6P.

Table 5: FLOATBench first fore-aft natural frequency (floating configuration) per tower (Hz); percentage change relative to REF in parentheses.

| | REF | OPT1 | OPT2 |
|---|---|---|---|
| FA1 (Hz) | 0.336 | 0.573 (+70.5 %) | 0.537 (+59.8 %) |

### E.2 SCHEMA REFERENCE

Table 6 documents the schema of the released CSVs. For each tower (REF, OPT1, OPT2) we release three files: `train_damage.csv` (51,840 rows, 18 columns), `test_damage.csv` (142,200 rows, 18 columns), and `data.csv` (194,040 rows, 16 columns: the raw rows without split or regime labels, kept as the canonical source for reproducing the partition). All three files carry three grid-coordinate columns (`wind_speed_id`, `wave_hs_id`, `wave_tp_id`) that index each row's position on the $22 \times 7 \times 7$ envelope; the deterministic train/test split is fully recovered by filtering on these IDs (Section F.1). The regime-label columns (`wind_group`, `wave_group`) appear only in `train_damage.csv` and `test_damage.csv` and are populated by the regime-aware splitter (Section 4.1): on train rows they are trivially `In-train`, and on test rows they hold the actual IT/IP/EX regime; the joint label `wind_wave_group` can be derived at load time as `wind_group + "_" + wave_group`.

Table 6: FLOATBench released CSV schema; one folder is released per tower (REF, OPT1, OPT2), so no tower identifier column is needed. Columns are grouped by category. Grid-coordinate IDs identify each row's position on the $22 \times 7 \times 7$ envelope and appear in all three files. Regime-label columns are `In-train` on train rows and carry the IT/IP/EX regime on test rows; they appear only in `train_damage.csv`/`test_damage.csv`.

| Column | Type | Description |
|---|---|---|
| *Identifiers* | | |
| `sim_id` | int | Unique simulation identifier (ties together the 30 sections of one run) |
| `section_id` | int | Tower section index $\in \{1, \ldots, 30\}$, 1 (base) to 30 (top) |
| `wind_speed_id` | int | Grid index $\in \{1, \ldots, 22\}$, ordered by `wind_speed` ascending |
| `wave_hs_id` | int | Grid index $\in \{1, \ldots, 7\}$ within each `wind_speed` |
| `wave_tp_id` | int | Grid index $\in \{1, \ldots, 7\}$ within each (`wind_speed`, `wave_hs`) |
| `wind_seed_id` | int | Turbulence seed index $\in \{1, \ldots, 6\}$ |
| *Environmental features* | | |
| `wind_speed` | float | Nominal hub-height wind speed (m/s) |
| `mean_wind_speed` | float | Realised 10-min mean hub-height wind speed (m/s) |
| `std_wind_speed` | float | Realised 10-min std of hub-height wind speed (m/s) |
| `wave_hs` | float | Significant wave height (m) |
| `wave_tp` | float | Wave peak period (s) |
| *Tower section geometry* | | |
| `section_height_m` | float | Tower section midpoint height along tower axis (m) |
| `section_radius_m` | float | Tower section outer radius (m) |
| `section_thickness_m` | float | Tower section wall thickness (m) |
| *Regime labels* (only in `train_damage.csv`/`test_damage.csv`) | | |
| `wind_group` | str | `In-train` on train rows; IT/IP/EX on test rows (wind axis) |
| `wave_group` | str | `In-train` on train rows; IT/IP/EX on test rows (wave axis) |
| *Damage targets* | | |
| `damage` | float | Miner-summed fatigue damage at the section (dimensionless) |
| `damage_weight` | float | Expected number of 600 s simulation windows the simulation represents over the 25-year service life (occurrence probability $\times$ 1,314,000 lifetime windows); lifetime damage is $\sum_i \texttt{damage}_i \cdot \texttt{damage_weight}_i$ over all conditions |

**Lifetime weighting (`damage_weight`).** Each simulation's weight is the expected number of times its 600 s window occurs over the 25-year service life: $\texttt{damage_weight} = P \times 1{,}314{,}000$, where $P$ is the occurrence probability of the simulated state and $1{,}314{,}000$ is the number of 600 s windows in 25 years. Wind is kept in discrete probability bins and the 49 wave states and 6 seeds of a wind level carry equal mass (Alves Ribeiro et al., 2026), so a simulation at wind level $v$ has weight $1{,}314{,}000 \times P(v)/294$ (22 unique values per tower; the weights of a tower sum to $1{,}314{,}000$). Lifetime damage per section is $D_{\text{life}} = \sum_i \texttt{damage_weight}_i \, D_i$. The weight enters no benchmark metric; it is only used to aggregate lifetime damage. The released value is the normalized wind-bin probability times the $1{,}314{,}000$ lifetime windows, divided equally over the 294 simulations of that wind level. Because every per-condition label is released, users can re-weight the states for another site whose met-ocean distribution is covered by the sampled envelope.

# F BENCHMARK DETAILS

## F.1 TRAIN/TEST SELECTION (REGIME-LABEL REFERENCE)

The train/test partition is defined by a deterministic preset on the $22 \times 7 \times 7$ envelope; Table 7 lists the train and test IDs per variable.

Table 7: FLOATBench train/test partition: train and test IDs per variable.

| Variable | Train IDs | Test IDs | Train count | Test count |
|---|---|---|---|---|
| Wind speed, $V$ (22 levels) | $\{2\text{–}7,\ 9\text{–}14,\ 16\text{–}21\}$ | $\{1, 8, 15, 22\}$ | 18 | 4 |
| Wave height, $H_s$ (7 levels) | $\{2, 3, 5, 6\}$ | $\{1, 4, 7\}$ | 4 | 3 |
| Wave peak period, $T_p$ (7 levels) | $\{2, 3, 5, 6\}$ | $\{1, 4, 7\}$ | 4 | 3 |

## F.2 REGIME-LABELING ALGORITHM AND DISTANCE DISTRIBUTIONS

**Algorithm.** Algorithm 1 formalizes the three-stage regime labeling described in Section 4.1, applied independently on the wind $(\mathrm{mean}\, V, \mathrm{std}\, V)$ and wave $(H_s, T_p)$ subspaces.

**Algorithm 1** REGIMELABEL: distance + alpha-shape regime labeling.

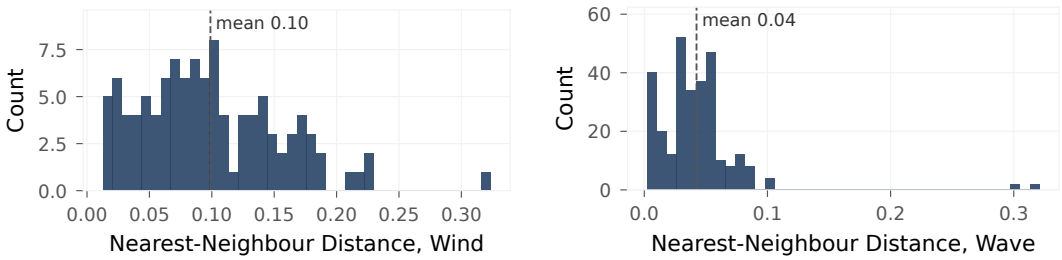
**Require:** Training points $X_{\mathrm{tr}}$, test points $X_{\mathrm{te}}$ (per subspace), threshold $\tau = 0.5$, hull parameter $\alpha = 0.1$
**Ensure:** Labels $L \in \{\mathrm{IT}, \mathrm{IP}, \mathrm{EX}\}^{|X_{\mathrm{te}}|}$ (In-train, Interpolation, Extrapolation)

**Stage 1 (train→train spacing)**
**for** each $x \in X_{\mathrm{tr}}^{\mathrm{unique}}$ **do**
  $d_x \leftarrow \min_{y \in X_{\mathrm{tr}},\, y \neq x} \|x - y\|$ ▷ standardized units
**end for**
$s \leftarrow \mathrm{mean}(d_x)$ ▷ spacing scale; see Figure 9, Table 8
**Stage 2 (test→train, distance-based grouping)**
**for** each $x \in X_{\mathrm{te}}$ **do**
  $\hat{d}_x \leftarrow \min_{y \in X_{\mathrm{tr}}} \|x - y\| / s$
  **if** $\hat{d}_x \leq \tau$ **then**
    $L_x \leftarrow$ IT
  **else**
    $L_x \leftarrow$ IP
  **end if**
**end for**
**Stage 3 (boundary-based override)**
$H \leftarrow \text{alpha-shape}(X_{\mathrm{tr}};\, \alpha)$ ▷ original units
$\varepsilon \leftarrow \tau\, s$ ▷ $s$ standardized; $\varepsilon$ applied to $\mathrm{dist}(x, \partial H)$ in original units
**for** each $x \in X_{\mathrm{te}}$ **do**
  **if** $x \notin H$ **and** $\mathrm{dist}(x, \partial H) \geq \varepsilon$ **then**
    $L_x \leftarrow$ EX
  **end if**
**end for**
**return** $L$

**Distance histograms and threshold derivation.** Figure 9 and Table 8 summarize the train-spacing statistics; $\tau = 0.5$ places the IT/IP boundary at half the mean train-train spacing per subspace.

Table 8: Train-spacing summary on the standardized wind and wave subspaces. The *mean* column is the spacing scale used to normalize test-to-train distances.

| subspace | $n$ unique train pts | mean | median | p25 | p90 |
|---|---|---|---|---|---|
| wind | 108 | 0.099 | 0.089 | 0.056 | 0.171 |
| wave | 288 | 0.042 | 0.036 | 0.026 | 0.073 |




(a) Wind subspace (b) Wave subspace

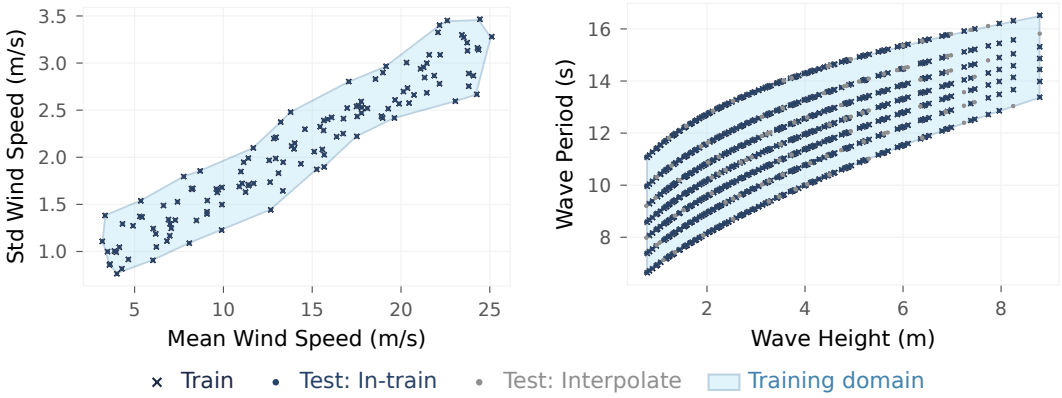
Figure 9: Train-train nearest-neighbor spacing on the standardized wind and wave subspaces. The dashed line marks the mean (spacing scale).

### F.3 Random vs regime-aware split: regime composition

Table 9 contrasts the regime composition of the test set under the random and regime-aware partitions; both use 142,200 test rows per tower. Under the random partition (Figure 10), training samples cover the full envelope, so the alpha-shape hull covers the entire envelope: no test point lies outside the hull, every EX regime is empty, and the test set collapses to IT_IT (86%) and IT_IP (14%). Under the regime-aware partition, training is restricted to the inner envelope.

Table 9: Test-set composition by regime, under random vs regime-aware partitions.

| | Random | | Regime-aware | |
|---|---|---|---|---|
| Regime cell (wind_wave) | rows | % | rows | % |
| IT_IT | 122,760 | 86.33 | 22,560 | 15.86 |
| IT_IP | 19,440 | 13.67 | 37,350 | 26.27 |
| IT_EX | 0 | 0.00 | 51,420 | 36.16 |
| IP_IT | 0 | 0.00 | 4,920 | 3.46 |
| IP_IP | 0 | 0.00 | 5,280 | 3.71 |
| IP_EX | 0 | 0.00 | 4,500 | 3.16 |
| EX_IT | 0 | 0.00 | 2,760 | 1.94 |
| EX_IP | 0 | 0.00 | 5,790 | 4.07 |
| **EX_EX** | **0** | **0.00** | **7,620** | **5.36** |
| Total | 142,200 | 100 | 142,200 | 100 |




(a) Wind subspace (b) Wave subspace

Figure 10: Random validation regime partition; the alpha-shape hull is shaded. All test points fall in IT/IP regimes, with no EX samples.

### F.4 BASELINE MODELS

We describe the eight surrogate families evaluated. Per-cell counts of configurations actually trained are in Table 10.

**XGBoost (Chen & Guestrin, 2016).** Gradient-boosted decision trees with histogram splits and second-order gradient information; portfolio configurations vary tree depth, learning rate, and regularisation.

**LightGBM (Ke et al., 2017).** Histogram-based gradient boosting with leaf-wise tree growth; portfolio configurations vary leaves, learning rate, and feature/row sub-sampling.

**CatBoost (Prokhorenkova et al., 2018).** Gradient boosting with oblivious (symmetric) trees and ordered boosting to mitigate prediction shift; portfolio configurations vary depth, learning rate, and regularisation.

**NeuralNetFastAI (Howard & Gugger, 2020).** Tabular MLP from the fastai library with 1-cycle learning-rate schedule and embedded categorical features; portfolio configurations vary layer widths, dropout, learning rate, and weight decay.

**NeuralNetTorch (Erickson et al., 2020).** AutoGluon's PyTorch tabular MLP with configurable depth, width, dropout, and categorical embeddings, trained with Adam and early stopping.

**TabM (Gorishniy et al., 2025).** Parameter-efficient ensemble of $k$ MLPs that share most weights (BatchEnsemble-style member adapters) and whose predictions are averaged; AutoGluon uses the TabM-mini variant with $k = 32$, available under AutoGluon's `extreme` preset.

**ExtraTrees (Geurts et al., 2006).** Trees grown on the full training sample, with split thresholds drawn at random and the best of $K$ random candidate splits kept per node; portfolio configurations vary `max_features`, `max_leaf_nodes`, and `min_samples_leaf`.

**RandomForest (Breiman, 2001).** Bagged decision trees with bootstrap row sampling and random feature subsets per split; portfolio configurations vary `max_features`, `max_leaf_nodes`, and `min_samples_leaf` (e.g., `RandomForestMSE`, `RandomForest_r195`).

**Stacked ensembles via AutoGluon.** The eight families above are trained as independent base learners through AutoGluon Tabular (Erickson et al., 2020), which combines $k$-fold bagging with multi-layer stacking and builds its `WeightedEnsemble` models by greedy forward ensemble selection with replacement (Caruana et al., 2004). AutoGluon stacks models in three layers: base learners (L1), level-2 ensembles (the global weighted ensemble WE_L2 plus per-family BAG_L2 stackers), and an optional level-3 weighted ensemble (WE_L3).

- **L1** (base learners): each family fits one or more configurations under 8-fold bagging, named `<Family>_BAG_L1` (e.g., `CatBoost_BAG_L1`, `NeuralNetFastAI_r102_BAG_L1`); held-out fold predictions are stored.
- **WE_L2** (greedy weighted ensemble of L1): sparse non-negative weights fitted on the L1 predictions for the 20% validation hold-out; the canonical ensemble baseline.
- **WE_L3** (weighted ensemble on top of L2 + L1): a further weighted ensemble over L2 stacker outputs together with L1 predictions.
- **BAG_L2** stackers: alternative L2 layer where each base family is refit on L1 held-out predictions plus original features, named `<Family>_BAG_L2` (e.g., `NeuralNetFastAI_BAG_L2`); distinct from the WE_L{2,3} weighted ensembles.

**Foundation model exclusion.** AutoGluon's `extreme` preset nominally includes TabPFN-v2 (Hollmann et al., 2025), TabICL (Qu et al., 2025), and Mitra (Zhang et al., 2025), but none trains on this regression task. TabICL supports classification only in AutoGluon and is skipped. AutoGluon caps TabPFN-v2 and Mitra at 10,000 training rows; FLOATBench's $\approx$41,500-row training fold (51,840

rows minus the 20% internal validation hold-out, Section F.5) is $\approx 4\times$ the cap, and both abort before training.

Table 10: Trained models per (experiment, tower or fold, preset), by family.

| Exp. | Split | Preset | XGB | LGBM | CB | NN-F | NN-T | TabM | ET | RF | WE | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E1 | REF | best | 8 | 15 | 14 | 13 | 12 | 0 | 6 | 4 | 2 | 74 |
| | | extreme | 2 | 3 | 5 | 0 | 0 | 6 | 0 | 0 | 1 | 17 |
| | | **Total** | **10** | **18** | **19** | **13** | **12** | **6** | **6** | **4** | **3** | **91** |
| | OPT1 | best | 8 | 17 | 15 | 14 | 12 | 0 | 6 | 4 | 2 | 78 |
| | | extreme | 2 | 3 | 5 | 0 | 0 | 6 | 0 | 0 | 1 | 17 |
| | | **Total** | **10** | **20** | **20** | **14** | **12** | **6** | **6** | **4** | **3** | **95** |
| | OPT2 | best | 7 | 15 | 12 | 10 | 10 | 0 | 5 | 4 | 2 | 65 |
| | | extreme | 2 | 3 | 5 | 0 | 0 | 6 | 0 | 0 | 1 | 17 |
| | | **Total** | **9** | **18** | **17** | **10** | **10** | **6** | **5** | **4** | **3** | **82** |
| E2 | REF | best | 8 | 16 | 15 | 15 | 12 | 0 | 6 | 4 | 2 | 78 |
| | | extreme | 2 | 3 | 5 | 0 | 0 | 6 | 0 | 0 | 1 | 17 |
| | | **Total** | **10** | **19** | **20** | **15** | **12** | **6** | **6** | **4** | **3** | **95** |
| | OPT1 | best | 8 | 17 | 15 | 15 | 12 | 0 | 6 | 4 | 2 | 79 |
| | | extreme | 2 | 3 | 5 | 0 | 0 | 6 | 0 | 0 | 1 | 17 |
| | | **Total** | **10** | **20** | **20** | **15** | **12** | **6** | **6** | **4** | **3** | **96** |
| | OPT2 | best | 8 | 16 | 15 | 15 | 12 | 0 | 6 | 4 | 2 | 78 |
| | | extreme | 2 | 3 | 5 | 0 | 0 | 6 | 0 | 0 | 1 | 17 |
| | | **Total** | **10** | **19** | **20** | **15** | **12** | **6** | **6** | **4** | **3** | **95** |
| E3 | REF+OPT1 | best | 5 | 12 | 10 | 7 | 7 | 0 | 4 | 3 | 2 | 50 |
| | | extreme | 1 | 2 | 4 | 0 | 0 | 5 | 0 | 0 | 1 | 13 |
| | | **Total** | **6** | **14** | **14** | **7** | **7** | **5** | **4** | **3** | **3** | **63** |
| | REF+OPT2 | best | 4 | 11 | 10 | 7 | 6 | 0 | 4 | 3 | 2 | 47 |
| | | extreme | 1 | 2 | 4 | 0 | 0 | 5 | 0 | 0 | 1 | 13 |
| | | **Total** | **5** | **13** | **14** | **7** | **6** | **5** | **4** | **3** | **3** | **60** |
| | OPT1+OPT2 | best | 4 | 11 | 9 | 6 | 6 | 0 | 4 | 3 | 2 | 45 |
| | | extreme | 1 | 2 | 4 | 0 | 0 | 5 | 0 | 0 | 1 | 13 |
| | | **Total** | **5** | **13** | **13** | **6** | **6** | **5** | **4** | **3** | **3** | **58** |

*Columns:* XGB = XGBoost; LGBM = LightGBM (incl. LightGBMXT, LightGBMLarge); CB = CatBoost; NN-F = NeuralNetFastAI; NN-T = NeuralNetTorch; ET = ExtraTrees (incl. ExtraTreesMSE); RF = RandomForest (incl. RandomForestMSE); WE = WeightedEnsemble.

### F.5 TRAINING CONFIGURATION

**Validation split.** When no explicit validation CSV is provided, the trainer carves a $20\%$ holdout from the training partition, grouped by simulation: every row of a given simulation lands entirely in either the training or validation fold. This avoids the leakage that would otherwise arise because the 30 per-section rows of one simulation share the same wind/wave conditions and are strongly correlated.

**Compute.** AutoGluon runs were trained on the local machine described in Appendix A. Each (experiment, tower or fold, preset) cell received 24 vCPUs and 1 GPU (12 vCPUs and 1 shared GPU per bagging fold, 8 folds) under a 4-hour wall-clock budget. The 18 cells (E1, E2, E3 across 3 towers or 3 cross-tower folds, each under `best` and `extreme`) total at most 72 hours of wall-clock time.

### F.6 METRIC DEFINITIONS

To rigorously evaluate surrogate predictions $\hat{y}_i$ against ground-truth DEL values $y_i$ ($N$ test points), we employ a set of well-established metrics that quantify accuracy, error, and generalization. These metrics are defined as follows.

**Mean Absolute Error (MAE).** The average magnitude of absolute errors between predictions and ground truth,

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|.$$

This metric provides an intuitive measure of the average error magnitude, making it easy to interpret.

**Mean Squared Error (MSE).** The average squared difference between predictions and ground truth,

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2.$$

Penalizes larger deviations more severely, emphasizing outliers and high-error regimes. Common as a regression training loss.

**Root Mean Square Error (RMSE).** The square root of MSE,

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}.$$

Penalizes large errors more heavily than MAE due to the squaring operation, making it sensitive to outliers.

**Coefficient of Determination ($R^2$).** The proportion of variance in the ground truth explained by the model,

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2},$$

where $\bar{y}$ is the mean of $y_i$. A value close to 1 indicates that the model explains most of the variance; values close to 0 (or negative) suggest poor predictive performance.

**Relative $L^2$ Error.** The Euclidean-norm error normalized by the ground-truth norm,

$$\mathrm{Rel}\, L^2 = \frac{\|\hat{y} - y\|_2}{\|y\|_2}.$$

Scale-invariant; particularly useful for comparing models across regimes of different magnitudes. Used as the cross-model anchor in the leaderboards.

**Mean Relative Error (MRE).** The point-wise mean of relative deviations,

$$\mathrm{MRE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y}_i - y_i|}{|y_i|}.$$

Each test point contributes equally, preserving the regime hierarchy under skewed magnitude distributions. Reported for per-regime diagnostics.

**Maximum Error (MaxErr).** The largest absolute error across the test set,

$$\mathrm{MaxErr} = \max_i |\hat{y}_i - y_i|.$$

Captures worst-case performance, critical in fatigue applications where large prediction errors can have safety consequences.

These metrics collectively capture absolute and relative errors, variance explained, and worst-case behavior, providing a comprehensive evaluation framework. Lower is better for every metric except $R^2$.

### F.7 BOOTSTRAP CONFIDENCE INTERVALS

To quantify the statistical uncertainty of the leaderboard metrics, we use a condition-level (cluster) bootstrap (Efron, 1979; Field & Welsh, 2007) with percentile intervals (Efron, 1981). This nonparametric approach makes no normality assumption.

**Bootstrap methodology.** Given $N$ test pairs $\{(\hat{y}_i, y_i)\}_{i=1}^{N}$, grouped into $K$ operating conditions (wind level, $H_s$, $T_p$), we draw $B$ replicates by sampling $K$ conditions with replacement and keeping all rows of each drawn condition (its 6 seeds and 30 sections); $I^{(b)}$ is the resulting row multiset. Whole conditions are resampled because the rows of one condition share its met-ocean state: row-level resampling gives standard deviations of Rel $L^2$ DEL smaller by a median $4.8\times$ (E1), $8.3\times$ (E2), and $4.2\times$ (E3). For each replicate $b$ we compute the metric $\theta^{(b)} = \mathcal{M}\big(\{(\hat{y}_i, y_i) : i \in I^{(b)}\}\big)$. The bootstrap estimates of the mean and standard deviation are

$$\bar{\theta}_{\text{boot}} = \frac{1}{B}\sum_{b=1}^{B}\theta^{(b)}, \qquad \sigma_{\text{boot}} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\big(\theta^{(b)} - \bar{\theta}_{\text{boot}}\big)^2},$$

and the percentile-method confidence interval is $\text{CI}_{1-\alpha} = [\theta^*_{\alpha/2}, \theta^*_{1-\alpha/2}]$, where $\theta^*_p$ is the $p$-th percentile of the bootstrap distribution.

**Implementation details.** We draw $B = 2{,}000$ resamples at the 95% confidence level ($\alpha = 0.05$) with RNG seed $s = 42$; the eight metrics (Section F.6) share one bootstrap pass (Algorithm 2).

**Algorithm 2** BOOTSTRAPREGRESSIONMETRICS: percentile bootstrap for the leaderboard metrics.

**Require:** Predictions and ground truth $\{(\hat{y}_i, y_i)\}_{i=1}^{N}$ with condition labels $k_i \in \{1, \ldots, K\}$, metric set $\mathcal{M} = \{f_m\}_{m=1}^{M}$, resamples $B$ (default 2000), significance level $\alpha$ (default 0.05), RNG seed $s$ (default 42)
**Ensure:** For each metric $m$: bootstrap mean $\bar{\theta}_m$, standard deviation $\sigma_m$, percentile CI $[\theta^*_{m,\alpha/2}, \theta^*_{m,1-\alpha/2}]$

Allocate $\Theta_m \in \mathbb{R}^B$ for every $m \in \mathcal{M}$ and seed RNG with $s$
**for** $b = 1, \ldots, B$ **do**
  Sample $K$ conditions with replacement, $c^{(b)} \sim \text{Uniform}\{1, \ldots, K\}^K$; $I^{(b)} \leftarrow$ all rows of the conditions in $c^{(b)}$
  $(\hat{y}^{(b)}, y^{(b)}) \leftarrow (\hat{y}_{I^{(b)}},\, y_{I^{(b)}})$
  **for** $m = 1, \ldots, M$ **do**
    $\Theta_m[b] \leftarrow f_m(\hat{y}^{(b)}, y^{(b)})$ ▷ shared intermediates ($\hat{y} - y$, $|\hat{y} - y|$, $\|y\|$) reused across $m$
  **end for**
**end for**
**for** $m = 1, \ldots, M$ **do**
  $\bar{\theta}_m \leftarrow \frac{1}{B}\sum_{b=1}^{B}\Theta_m[b]$
  $\sigma_m \leftarrow \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\Theta_m[b] - \bar{\theta}_m)^2}$ ▷ ddof $= 1$
  $\theta^*_{m,\alpha/2},\, \theta^*_{m,1-\alpha/2} \leftarrow \text{percentile}_{\text{linear}}(\Theta_m;\, \alpha/2,\, 1-\alpha/2)$
**end for**
**return** $\{(\bar{\theta}_m,\, \sigma_m,\, [\theta^*_{m,\alpha/2},\, \theta^*_{m,1-\alpha/2}])\}_{m=1}^{M}$

**Interpretation.** The CIs separate genuine model gaps from finite-sample noise without normality assumptions. Overlapping 95% CIs mean that a gap may not be significant; because all models share the same condition resamples, the distribution of their paired difference gives a sharper test than marginal overlap. Wide CIs flag sensitivity to test-set composition. A worked example is shown in Table 11.

Table 11: Worked example (two hypothetical surrogates $S_1$, $S_2$): overlapping vs. disjoint 95% CIs.

| Case | $S_1$: $R^2$, 95% CI | $S_2$: $R^2$, 95% CI | Verdict |
|---|---|---|---|
| A | 0.950, [0.940, 0.960] | 0.945, [0.929, 0.961] | CIs overlap; not distinguishable |
| B | 0.950, [0.946, 0.954] | 0.930, [0.926, 0.934] | CIs disjoint; $S_1$ significantly better |

# G LEADERBOARDS

This appendix collects the top-10 leaderboards used in the main text. All three tables show model, preset, MRE, $R^2$ and Rel $L^2$ DEL (auxiliary error and compute metrics are in the released leaderboard CSVs) and are sorted by Rel $L^2$ DEL.

## G.1 TOP-10 SURROGATES UNDER RANDOM (E1), PER TOWER

Table 12 covers the E1 random partition, per tower.

## G.2 TOP-10 SURROGATES UNDER REGIME-AWARE (E2), PER TOWER

Table 13 covers the E2 regime-aware partition, per tower.

## G.3 TOP-10 SURROGATES UNDER CROSS-TOWER (E3), PER FOLD

Table 14 covers the E3 cross-tower transfer protocol, per leave-one-tower-out fold.

Table 12: **Top-10 random (E1) per tower, sorted by Rel L$^2$ DEL.** Error metrics on the DEL target; cells are $\bar{\theta}_{\text{boot}} \pm \sigma_{\text{boot}}$ (bootstrap mean $\pm$ bootstrap standard deviation, $B = 2{,}000$ resamples of whole operating conditions, Appendix F.7). Full $95\%$ percentile-bootstrap intervals are computed by Algorithm 2 and released alongside the leaderboard CSVs. Columns shown: model, preset, MRE, $R^2$ and Rel L$^2$; MSE, MAE, RMSE, MaxErr, latency, throughput and training time are in the released leaderboard CSVs.

| Rk | Model | Preset | MRE (%) | $R^2$ | Rel L$^2$ |
|---|---|---|---|---|---|
| REF | | | | | |
| 1 | WeightedEnsemble_L2 | extreme | $2.289 \pm 0.077$ | $0.998109 \pm 0.000071$ | $0.01909 \pm 0.00028$ |
| 2 | TabM_r191_BAG_L1 | extreme | $2.89 \pm 0.14$ | $0.997934 \pm 0.000069$ | $0.01995 \pm 0.00031$ |
| 3 | CatBoost_r177_BAG_L1 | best | $2.136 \pm 0.049$ | $0.997915 \pm 0.000080$ | $0.02005 \pm 0.00027$ |
| 4 | WeightedEnsemble_L2 | best | $2.104 \pm 0.049$ | $0.997913 \pm 0.000080$ | $0.02005 \pm 0.00027$ |
| 5 | CatBoost_r69_BAG_L1 | best | $2.149 \pm 0.050$ | $0.997913 \pm 0.000079$ | $0.02005 \pm 0.00027$ |
| 6 | WeightedEnsemble_L3 | best | $2.078 \pm 0.047$ | $0.997906 \pm 0.000080$ | $0.02009 \pm 0.00027$ |
| 7 | CatBoost_BAG_L1 | best | $2.201 \pm 0.050$ | $0.997868 \pm 0.000081$ | $0.02027 \pm 0.00027$ |
| 8 | TabM_BAG_L1 | extreme | $2.220 \pm 0.056$ | $0.997856 \pm 0.000084$ | $0.02032 \pm 0.00030$ |
| 9 | CatBoost_BAG_L1 | extreme | $2.221 \pm 0.051$ | $0.997843 \pm 0.000081$ | $0.02039 \pm 0.00027$ |
| 10 | CatBoost_r91_BAG_L1 | extreme | $2.141 \pm 0.050$ | $0.997842 \pm 0.000083$ | $0.02039 \pm 0.00028$ |
| OPT1 | | | | | |
| 1 | WeightedEnsemble_L2 | extreme | $3.501 \pm 0.059$ | $0.98928 \pm 0.00051$ | $0.04064 \pm 0.00056$ |
| 2 | CatBoost_BAG_L1 | best | $3.506 \pm 0.053$ | $0.98927 \pm 0.00051$ | $0.04065 \pm 0.00057$ |
| 3 | CatBoost_BAG_L1 | extreme | $3.517 \pm 0.053$ | $0.98927 \pm 0.00051$ | $0.04066 \pm 0.00057$ |
| 4 | WeightedEnsemble_L2 | best | $3.456 \pm 0.051$ | $0.98923 \pm 0.00052$ | $0.04074 \pm 0.00058$ |
| 5 | WeightedEnsemble_L3 | best | $3.456 \pm 0.051$ | $0.98923 \pm 0.00052$ | $0.04074 \pm 0.00058$ |
| 6 | CatBoost_r177_BAG_L1 | best | $3.458 \pm 0.051$ | $0.98921 \pm 0.00052$ | $0.04078 \pm 0.00058$ |
| 7 | CatBoost_r137_BAG_L1 | best | $3.618 \pm 0.055$ | $0.98905 \pm 0.00052$ | $0.04107 \pm 0.00058$ |
| 8 | CatBoost_r13_BAG_L1 | best | $3.718 \pm 0.057$ | $0.98883 \pm 0.00052$ | $0.04149 \pm 0.00053$ |
| 9 | CatBoost_r51_BAG_L1 | extreme | $3.627 \pm 0.053$ | $0.98866 \pm 0.00053$ | $0.04180 \pm 0.00054$ |
| 10 | CatBoost_r69_BAG_L1 | best | $3.828 \pm 0.057$ | $0.98844 \pm 0.00053$ | $0.04221 \pm 0.00052$ |
| OPT2 | | | | | |
| 1 | WeightedEnsemble_L2 | extreme | $3.513 \pm 0.069$ | $0.98859 \pm 0.00058$ | $0.04049 \pm 0.00060$ |
| 2 | CatBoost_BAG_L1 | best | $3.477 \pm 0.054$ | $0.98839 \pm 0.00058$ | $0.04084 \pm 0.00058$ |
| 3 | CatBoost_BAG_L1 | extreme | $3.474 \pm 0.055$ | $0.98838 \pm 0.00058$ | $0.04085 \pm 0.00058$ |
| 4 | CatBoost_r177_BAG_L1 | best | $3.459 \pm 0.054$ | $0.98836 \pm 0.00058$ | $0.04089 \pm 0.00058$ |
| 5 | CatBoost_r137_BAG_L1 | best | $3.586 \pm 0.055$ | $0.98819 \pm 0.00058$ | $0.04119 \pm 0.00058$ |
| 6 | WeightedEnsemble_L3 | best | $3.507 \pm 0.053$ | $0.98815 \pm 0.00059$ | $0.04125 \pm 0.00059$ |
| 7 | WeightedEnsemble_L2 | best | $3.507 \pm 0.053$ | $0.98815 \pm 0.00059$ | $0.04125 \pm 0.00059$ |
| 8 | CatBoost_r13_BAG_L1 | best | $3.668 \pm 0.058$ | $0.98805 \pm 0.00058$ | $0.04143 \pm 0.00055$ |
| 9 | CatBoost_r51_BAG_L1 | extreme | $3.609 \pm 0.055$ | $0.98782 \pm 0.00060$ | $0.04183 \pm 0.00056$ |
| 10 | CatBoost_r50_BAG_L1 | best | $3.539 \pm 0.054$ | $0.98761 \pm 0.00062$ | $0.04219 \pm 0.00060$ |

Table 13: **Top-10 regime-aware (E2) per tower, sorted by Rel $L^2$ DEL.** Error metrics on the DEL target; cells are $\bar{\theta}_{\text{boot}} \pm \sigma_{\text{boot}}$ (bootstrap mean $\pm$ bootstrap standard deviation, $B = 2{,}000$ resamples of whole operating conditions, Appendix F.7). Full $95\%$ percentile-bootstrap intervals are computed by Algorithm 2 and released alongside the leaderboard CSVs; top-3 Rel $L^2$ bounds are reproduced in Table 16. Columns shown: model, preset, MRE, $R^2$ and Rel $L^2$; MSE, MAE, RMSE, MaxErr, latency, throughput and training time are in the released leaderboard CSVs.

| Rk | Model | Preset | MRE (%) | $R^2$ | Rel $L^2$ |
|---|---|---|---|---|---|
| REF | | | | | |
| 1 | WeightedEnsemble_L2 | best | $7.33 \pm 0.56$ | $0.9796 \pm 0.0013$ | $0.0634 \pm 0.0019$ |
| 2 | NeuralNetTorch_r22_BAG_L1 | best | $6.79 \pm 0.45$ | $0.9795 \pm 0.0014$ | $0.0635 \pm 0.0020$ |
| 3 | LightGBMLarge_BAG_L1 | best | $7.13 \pm 0.53$ | $0.9795 \pm 0.0013$ | $0.0636 \pm 0.0019$ |
| 4 | NeuralNetFastAI_r145_BAG_L2 | best | $7.12 \pm 0.53$ | $0.9794 \pm 0.0013$ | $0.0637 \pm 0.0019$ |
| 5 | NeuralNetFastAI_r11_BAG_L2 | best | $7.29 \pm 0.55$ | $0.9793 \pm 0.0013$ | $0.0638 \pm 0.0019$ |
| 6 | NeuralNetFastAI_r191_BAG_L2 | best | $7.08 \pm 0.53$ | $0.9793 \pm 0.0013$ | $0.0638 \pm 0.0019$ |
| 7 | LightGBM_r130_BAG_L2 | best | $7.06 \pm 0.53$ | $0.9792 \pm 0.0014$ | $0.0640 \pm 0.0019$ |
| 8 | XGBoost_r194_BAG_L2 | best | $7.08 \pm 0.53$ | $0.9792 \pm 0.0014$ | $0.0641 \pm 0.0020$ |
| 9 | NeuralNetFastAI_BAG_L2 | best | $7.36 \pm 0.56$ | $0.9791 \pm 0.0013$ | $0.0641 \pm 0.0019$ |
| 10 | XGBoost_BAG_L2 | best | $7.07 \pm 0.53$ | $0.9791 \pm 0.0014$ | $0.0641 \pm 0.0020$ |
| OPT1 | | | | | |
| 1 | WeightedEnsemble_L2 | best | $9.32 \pm 0.68$ | $0.9707 \pm 0.0016$ | $0.0692 \pm 0.0017$ |
| 2 | CatBoost_r13_BAG_L1 | best | $9.44 \pm 0.69$ | $0.9690 \pm 0.0016$ | $0.0712 \pm 0.0018$ |
| 3 | WeightedEnsemble_L3 | best | $9.36 \pm 0.68$ | $0.9689 \pm 0.0017$ | $0.0712 \pm 0.0018$ |
| 4 | CatBoost_r69_BAG_L1 | best | $9.36 \pm 0.67$ | $0.9686 \pm 0.0017$ | $0.0716 \pm 0.0018$ |
| 5 | CatBoost_BAG_L1 | extreme | $9.32 \pm 0.67$ | $0.9686 \pm 0.0017$ | $0.0716 \pm 0.0018$ |
| 6 | CatBoost_BAG_L1 | best | $9.34 \pm 0.67$ | $0.9685 \pm 0.0017$ | $0.0717 \pm 0.0018$ |
| 7 | LightGBM_r196_BAG_L2 | best | $9.22 \pm 0.62$ | $0.9683 \pm 0.0017$ | $0.0719 \pm 0.0017$ |
| 8 | CatBoost_r137_BAG_L1 | best | $9.44 \pm 0.67$ | $0.9681 \pm 0.0017$ | $0.0721 \pm 0.0018$ |
| 9 | NeuralNetFastAI_r11_BAG_L2 | best | $9.06 \pm 0.62$ | $0.9681 \pm 0.0018$ | $0.0722 \pm 0.0017$ |
| 10 | CatBoost_r177_BAG_L1 | best | $9.49 \pm 0.69$ | $0.9679 \pm 0.0017$ | $0.0724 \pm 0.0018$ |
| OPT2 | | | | | |
| 1 | WeightedEnsemble_L2 | best | $9.34 \pm 0.68$ | $0.9676 \pm 0.0017$ | $0.0704 \pm 0.0017$ |
| 2 | WeightedEnsemble_L3 | best | $9.43 \pm 0.69$ | $0.9662 \pm 0.0018$ | $0.0719 \pm 0.0018$ |
| 3 | CatBoost_r13_BAG_L1 | best | $9.51 \pm 0.70$ | $0.9662 \pm 0.0018$ | $0.0719 \pm 0.0018$ |
| 4 | CatBoost_r137_BAG_L1 | best | $9.45 \pm 0.67$ | $0.9654 \pm 0.0018$ | $0.0727 \pm 0.0017$ |
| 5 | CatBoost_r51_BAG_L1 | extreme | $9.49 \pm 0.69$ | $0.9654 \pm 0.0018$ | $0.0727 \pm 0.0017$ |
| 6 | CatBoost_r69_BAG_L1 | best | $9.52 \pm 0.69$ | $0.9651 \pm 0.0018$ | $0.0730 \pm 0.0018$ |
| 7 | CatBoost_r177_BAG_L1 | best | $9.64 \pm 0.71$ | $0.9647 \pm 0.0018$ | $0.0735 \pm 0.0018$ |
| 8 | NeuralNetTorch_BAG_L1 | best | $9.67 \pm 0.61$ | $0.9645 \pm 0.0019$ | $0.0737 \pm 0.0017$ |
| 9 | CatBoost_BAG_L1 | best | $9.66 \pm 0.71$ | $0.9643 \pm 0.0019$ | $0.0738 \pm 0.0018$ |
| 10 | CatBoost_BAG_L1 | extreme | $9.65 \pm 0.71$ | $0.9643 \pm 0.0018$ | $0.0738 \pm 0.0018$ |

Table 14: **Top-10 cross-tower (E3) per fold, sorted by Rel $L^2$ DEL.** Error metrics on the DEL target; cells are $\bar{\theta}_{\text{boot}} \pm \sigma_{\text{boot}}$ (bootstrap mean $\pm$ bootstrap standard deviation, $B = 2{,}000$ resamples of whole operating conditions, Appendix F.7). Full 95% percentile-bootstrap intervals are computed by Algorithm 2 and released alongside the leaderboard CSVs; top-3 Rel $L^2$ bounds are reproduced in Table 16. Columns shown: model, preset, MRE, $R^2$ and Rel $L^2$; MSE, MAE, RMSE, MaxErr, latency, throughput and training time are in the released leaderboard CSVs.

| Rk | Model | Preset | MRE (%) | $R^2$ | Rel $L^2$ |
|---|---|---|---|---|---|
| REF + OPT1 → OPT2 | | | | | |
| 1 | TabM_r52_BAG_L1 | extreme | $6.074 \pm 0.058$ | $0.9687 \pm 0.0012$ | $0.06700 \pm 0.00035$ |
| 2 | TabM_BAG_L1 | extreme | $8.05 \pm 0.16$ | $0.9545 \pm 0.0018$ | $0.08079 \pm 0.00036$ |
| 3 | NeuralNetFastAI_r102_BAG_L1 | best | $9.22 \pm 0.16$ | $0.9473 \pm 0.0022$ | $0.08696 \pm 0.00077$ |
| 4 | NeuralNetFastAI_r191_BAG_L1 | best | $9.44 \pm 0.19$ | $0.9457 \pm 0.0022$ | $0.08832 \pm 0.00061$ |
| 5 | TabM_r184_BAG_L1 | extreme | $8.495 \pm 0.044$ | $0.9433 \pm 0.0023$ | $0.09022 \pm 0.00037$ |
| 6 | NeuralNetFastAI_r145_BAG_L1 | best | $9.42 \pm 0.14$ | $0.9411 \pm 0.0024$ | $0.09191 \pm 0.00067$ |
| 7 | NeuralNetTorch_r22_BAG_L1 | best | $8.624 \pm 0.051$ | $0.9319 \pm 0.0028$ | $0.09887 \pm 0.00054$ |
| 8 | ExtraTreesMSE_BAG_L1 | best | $8.141 \pm 0.058$ | $0.9313 \pm 0.0031$ | $0.09933 \pm 0.00079$ |
| 9 | NeuralNetFastAI_BAG_L1 | best | $10.71 \pm 0.20$ | $0.9301 \pm 0.0033$ | $0.1001 \pm 0.0012$ |
| 10 | NeuralNetTorch_BAG_L1 | best | $11.51 \pm 0.28$ | $0.9226 \pm 0.0029$ | $0.10542 \pm 0.00069$ |
| REF + OPT2 → OPT1 | | | | | |
| 1 | TabM_r52_BAG_L1 | extreme | $10.254 \pm 0.084$ | $0.9382 \pm 0.0024$ | $0.09755 \pm 0.00043$ |
| 2 | NeuralNetFastAI_r191_BAG_L1 | best | $10.34 \pm 0.13$ | $0.9370 \pm 0.0024$ | $0.09848 \pm 0.00063$ |
| 3 | TabM_BAG_L1 | extreme | $11.49 \pm 0.10$ | $0.9222 \pm 0.0030$ | $0.10943 \pm 0.00051$ |
| 4 | NeuralNetFastAI_r102_BAG_L1 | best | $11.34 \pm 0.13$ | $0.9216 \pm 0.0034$ | $0.1098 \pm 0.0011$ |
| 5 | NeuralNetFastAI_BAG_L1 | best | $11.82 \pm 0.13$ | $0.9169 \pm 0.0034$ | $0.11309 \pm 0.00093$ |
| 6 | TabM_r69_BAG_L1 | extreme | $11.980 \pm 0.061$ | $0.9043 \pm 0.0037$ | $0.12139 \pm 0.00039$ |
| 7 | RandomForestMSE_BAG_L1 | best | $12.490 \pm 0.079$ | $0.9019 \pm 0.0037$ | $0.12289 \pm 0.00038$ |
| 8 | LightGBMXT_BAG_L1 | best | $9.472 \pm 0.080$ | $0.8981 \pm 0.0043$ | $0.12523 \pm 0.00087$ |
| 9 | XGBoost_BAG_L1 | best | $12.447 \pm 0.098$ | $0.8970 \pm 0.0039$ | $0.12589 \pm 0.00052$ |
| 10 | NeuralNetFastAI_r145_BAG_L1 | best | $14.08 \pm 0.27$ | $0.8961 \pm 0.0040$ | $0.12644 \pm 0.00095$ |
| OPT1 + OPT2 → REF | | | | | |
| 1 | NeuralNetFastAI_r191_BAG_L1 | best | $33.14 \pm 0.28$ | $0.070 \pm 0.032$ | $0.4227 \pm 0.0016$ |
| 2 | NeuralNetFastAI_BAG_L1 | best | $35.23 \pm 0.23$ | $0.023 \pm 0.033$ | $0.4331 \pm 0.0012$ |
| 3 | NeuralNetFastAI_r102_BAG_L1 | best | $41.04 \pm 0.24$ | $-0.251 \pm 0.043$ | $0.49029 \pm 0.00090$ |
| 4 | NeuralNetTorch_BAG_L1 | best | $47.64 \pm 0.13$ | $-0.509 \pm 0.050$ | $0.53841 \pm 0.00063$ |
| 5 | NeuralNetTorch_r79_BAG_L1 | best | $47.48 \pm 0.21$ | $-0.624 \pm 0.056$ | $0.5585 \pm 0.0010$ |
| 6 | NeuralNetFastAI_r145_BAG_L1 | best | $48.14 \pm 0.23$ | $-0.635 \pm 0.056$ | $0.56035 \pm 0.00091$ |
| 7 | NeuralNetTorch_r22_BAG_L1 | best | $51.62 \pm 0.17$ | $-0.684 \pm 0.057$ | $0.56875 \pm 0.00075$ |
| 8 | LightGBMXT_BAG_L1 | best | $54.26 \pm 0.17$ | $-0.829 \pm 0.062$ | $0.5927 \pm 0.0012$ |
| 9 | LightGBM_r188_BAG_L1 | best | $54.73 \pm 0.16$ | $-0.853 \pm 0.063$ | $0.5967 \pm 0.0012$ |
| 10 | TabM_r52_BAG_L1 | extreme | $49.89 \pm 0.18$ | $-0.870 \pm 0.064$ | $0.5993 \pm 0.0012$ |

### G.4 BOOTSTRAP CIS FOR TOP-3 SURROGATES (REL $L^2$, E2 AND E3)

The Rel $L^2$ DEL leaderboards in Table 13 and Table 14 report $\bar{\theta}_{\text{boot}} \pm \sigma_{\text{boot}}$ for column-budget reasons; Table 16 below adds the explicit 95% percentile-bootstrap interval $[\theta^*_{0.025}, \theta^*_{0.975}]$ for the top-3 surrogates of each E2 tower and each E3 fold, to make CI overlap directly inspectable. Full bounds for every metric and every rank are released alongside the leaderboard CSVs.

Under the condition-level bootstrap, adjacent ranks overlap in almost every E1 and E2 group, so near-tied surrogates should not be ordered from marginal intervals alone. Because all models of a group share the same condition resamples, we also test the paired rank-1 minus rank-2 difference in Rel $L^2$ DEL (Table 15). Rank 1 is significantly better in six of the nine groups; the top is tied on E2 REF, E1 OPT1, and the E3 fold REF+OPT2 → OPT1 (`TabM_r52_BAG_L1` vs. `NeuralNetFastAI_r191_BAG_L1`). No regime-level conclusion rests on these gaps: the rank reversal moves the Global rank-1 by 10 to 22 positions under the primary metric (Appendix H).

Table 15: **Paired rank-1 minus rank-2 difference in Rel $L^2$ DEL**: 95% interval of the condition-level bootstrap difference. An interval above zero means rank 1 is significantly better.

| Protocol | Group | Paired 95% interval | Verdict |
|---|---|---|---|
| E1 | REF | $[0.0006, 0.0011]$ | rank 1 better |
| E1 | OPT1 | $[-0.0004, 0.0005]$ | tied |
| E1 | OPT2 | $[0.00001, 0.0007]$ | rank 1 better |
| E2 | REF | $[-0.0008, 0.0010]$ | tied |
| E2 | OPT1 | $[0.0017, 0.0023]$ | rank 1 better |
| E2 | OPT2 | $[0.0013, 0.0018]$ | rank 1 better |
| E3 | REF+OPT1 → OPT2 | $[0.013, 0.014]$ | rank 1 better |
| E3 | REF+OPT2 → OPT1 | $[-0.0002, 0.0021]$ | tied |
| E3 | OPT1+OPT2 → REF | $[0.009, 0.011]$ | rank 1 better |

Table 16: **Rel $L^2$ DEL 95% percentile-bootstrap CIs for the top-3 surrogates**, per E2 tower and per E3 fold. Cells are $\bar{\theta}_{\text{boot}} \pm \sigma_{\text{boot}}$ $[\theta^*_{0.025}, \theta^*_{0.975}]$, with $B = 2{,}000$ condition-level resamples (Algorithm 2).

| Rank | Model | Rel $L^2$ DEL $\bar{\theta} \pm \sigma$ $[\theta^*_{0.025}, \theta^*_{0.975}]$ |
|---|---|---|
| *E2 (regime-aware), per tower* | | |
| REF | | |
| 1 | `WeightedEnsemble_L2` | $0.0634 \pm 0.0019$ $[0.0597, 0.0674]$ |
| 2 | `NeuralNetTorch_r22_BAG_L1` | $0.0635 \pm 0.0020$ $[0.0596, 0.0677]$ |
| 3 | `LightGBMLarge_BAG_L1` | $0.0636 \pm 0.0019$ $[0.0598, 0.0675]$ |
| OPT1 | | |
| 1 | `WeightedEnsemble_L2` | $0.0692 \pm 0.0017$ $[0.0659, 0.0726]$ |
| 2 | `CatBoost_r13_BAG_L1` | $0.0712 \pm 0.0018$ $[0.0678, 0.0747]$ |
| 3 | `WeightedEnsemble_L3` | $0.0712 \pm 0.0018$ $[0.0678, 0.0749]$ |
| OPT2 | | |
| 1 | `WeightedEnsemble_L2` | $0.0704 \pm 0.0017$ $[0.0670, 0.0738]$ |
| 2 | `WeightedEnsemble_L3` | $0.0719 \pm 0.0018$ $[0.0684, 0.0754]$ |
| 3 | `CatBoost_r13_BAG_L1` | $0.0719 \pm 0.0018$ $[0.0684, 0.0754]$ |
| *E3 (cross-tower), per fold* | | |
| REF+OPT1 → OPT2 | | |
| 1 | `TabM_r52_BAG_L1` | $0.06700 \pm 0.00035$ $[0.0663, 0.0677]$ |
| 2 | `TabM_BAG_L1` | $0.08079 \pm 0.00036$ $[0.0801, 0.0815]$ |
| 3 | `NeuralNetFastAI_r102_BAG_L1` | $0.08696 \pm 0.00077$ $[0.0855, 0.0884]$ |
| REF+OPT2 → OPT1 | | |
| 1 | `TabM_r52_BAG_L1` | $0.09755 \pm 0.00043$ $[0.0967, 0.0984]$ |
| 2 | `NeuralNetFastAI_r191_BAG_L1` | $0.09848 \pm 0.00063$ $[0.0973, 0.0997]$ |
| 3 | `TabM_BAG_L1` | $0.10943 \pm 0.00051$ $[0.1084, 0.1104]$ |
| OPT1+OPT2 → REF | | |
| 1 | `NeuralNetFastAI_r191_BAG_L1` | $0.4227 \pm 0.0016$ $[0.4196, 0.4259]$ |
| 2 | `NeuralNetFastAI_BAG_L1` | $0.4331 \pm 0.0012$ $[0.4308, 0.4354]$ |
| 3 | `NeuralNetFastAI_r102_BAG_L1` | $0.49029 \pm 0.00090$ $[0.4885, 0.4920]$ |

# H IS THE RANK REVERSAL AN ARTIFACT OF THE PARTITION?

The regime-aware partition (Section 4.1, Appendix F.2) has five fixed design choices: the alpha-shape parameter $\alpha$, the in-train distance threshold, the boundary tolerance, the train-spacing statistic, and the held-out grid. We test whether the labels and the rank reversal (the global rank-1 surrogate is not rank-1 in the joint-extrapolation cell EX_EX) depend on them.

**Why the first analysis is exact.** The first four parameters do not affect training. The train and test sets are fixed by the simulation grid (288 coarse-grid training conditions; the 790 held-out conditions are their complement, Appendix F.1). The parameters only relabel the test set, so every trained predictor is re-scored in one inference pass, without retraining. The relabeling reproduces the released labels at $100\%$ agreement. Recomputed per-regime $R^2$ matches the published leaderboards to below $6 \times 10^{-9}$, and Rel L$^2$ DEL to machine precision. Labels are identical on the three towers, since the partition uses only the shared wind/wave conditions.

## H.1 REGIME LABELS ARE STABLE UNDER THE PARTITION PARAMETERS

Labels are assigned per simulation, since the wind features are per-seed statistics: 790 conditions $\times$ 6 seeds $= 4{,}740$ labeled simulations. At the paper configuration the EX_EX cell holds $254$ simulations (62 conditions). We perturb one parameter at a time and report the fraction of simulations that keep their paper label, and the Jaccard index (intersection over union) of the EX_EX set with the paper's (Table 17, Figure 11).

Table 17: **Label stability of the regime-aware partition** (4,740 labeled test simulations; identical on all three towers). Baseline: $\alpha = 0.1$, threshold $0.5$, tolerance $\times 1$, mean spacing. The tolerance is varied as a multiplier of its released definition ($\varepsilon = \tau s$ with $s$ the standardized spacing scale, applied to the hull distance in original units: $0.049$ wind, $0.021$ wave; Section 4.1).

| Perturbation | Label agreement (%) | EX_EX Jaccard | EX_EX size (sims) |
|---|---|---|---|
| Baseline | 100.0 | 1.00 | 254 |
| $\alpha = 0.05$ ($\times 0.5$) | 98.97 | 0.94 | 240 |
| $\alpha = 0.075$ | 100.0 | 1.00 | 254 |
| $\alpha = 0.15, 0.2, 0.3$ (up to $\times 3$) | 97.93 | 0.90 | 282 |
| Threshold $0.25$ ($\times 0.5$) | 86.69 | 1.00 | 254 |
| Threshold $0.375$ | 93.73 | 1.00 | 254 |
| Threshold $0.75$ | 92.43 | 1.00 | 254 |
| Threshold $1.0$ ($\times 2$) | 83.50 | 1.00 | 254 |
| Tolerance $\times 0.5$, $\times 1.5$, $\times 2$ | 100.0 | 1.00 | 254 |
| Tolerance removed ($\times 0$) | 88.95 | 0.70 | 361 |
| Spacing statistic: mean $\rightarrow$ median | 96.18 | 1.00 | 254 |
| $\alpha = 10$ (falsification check) | 10.61 | 0.06 | 4,212 |

Labels are stable for $\alpha \in [0.05, 0.3]$ ($\times 0.5$ to $\times 3$): agreement stays at or above $97.9\%$ and the EX_EX Jaccard at or above $0.90$. The distance threshold moves only the In-train/Interpolate boundary. Halving or doubling it changes up to $16.5\%$ of the labels but never the EX_EX set, because the alpha-shape test alone decides extrapolation. The tolerance is conservative. Removing it grows EX_EX from $254$ to $361$ simulations; scaling it by $\times 0.5$ to $\times 2$ changes nothing on this grid. At $\alpha = 10$ the alpha shape degenerates and $89\%$ of the test set becomes EX_EX. This is the falsification check: the useful range of $\alpha$ is broad but bounded.

## H.2 THE RANK REVERSAL HOLDS IN EVERY VARIANT THAT KEEPS AN EXTRAPOLATION REGION

For each variant we recompute the EX_EX leaderboard of the full pool under Rel L$^2$ DEL and damage $R^2$ (Table 18).

The rank reversal holds in all 15 variants, on all towers, under both metrics. The EX_EX rank-1 is always a bagged `NeuralNetFastAI` model. At the paper configuration it is `NeuralNetFastAI_r102_BAG_L1` on all three towers. Under Rel L$^2$ DEL it stays rank-1

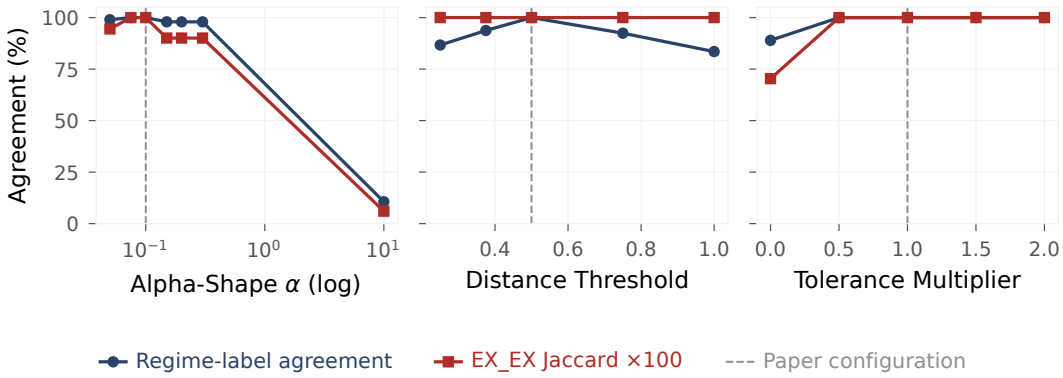



Figure 11: **Sensitivity of the regime labels to the partition parameters.** Regime-label agreement with the paper split and Jaccard index of the joint-extrapolation (EX_EX) set, as each parameter is perturbed around its paper value (dashed line). Left: alpha-shape parameter (log scale); center: in-train distance threshold; right: boundary-tolerance multiplier.

Table 18: **Conclusion stability over the 15 partition variants that keep an extrapolation region** (all rows of Table 17 except $\alpha = 10$). Ensemble = global rank-1 `WeightedEnsemble_L2`; EX_EX rank of the ensemble reported as baseline (range over variants). $\tau$ = Kendall $\tau$ between the variant's EX_EX ranking and the paper's. Pool: 95/96/95 models.

| Metric | Quantity | REF | OPT1 | OPT2 |
|---|---|---|---|---|
| Rel $L^2$ DEL | Rank reversal holds (variants) | 15/15 | 15/15 | 15/15 |
| | Ensemble EX_EX rank | 23 (18–23) | 11 (10–12) | 11 (11–13) |
| | EX_EX rank-1 | FastAI NN | FastAI NN | FastAI NN |
| | Min. $\tau$ vs. paper | 0.89 | 0.90 | 0.92 |
| Damage $R^2$ | Rank reversal holds (variants) | 15/15 | 15/15 | 15/15 |
| | Ensemble EX_EX rank | 10 (8–19) | 7 (6–7) | 8 (8) |
| | EX_EX rank-1 | FastAI NN | FastAI NN | FastAI NN |
| | Min. $\tau$ vs. paper | 0.79 | 0.976 | 0.98 |
| $\alpha = 10$ (degenerate): ensemble EX_EX rank, Rel $L^2$ / $R^2$ | | 2 / 1 | 1 / 1 | 1 / 1 |

in every variant except tolerance removal on REF, where `r156` takes rank 1. That least stable variant strengthens the finding: the ensemble falls to damage-$R^2$ EX_EX rank 19 ($0.946 \rightarrow 0.865$, $\tau = 0.79$). Excluding it, $\tau \geq 0.91$ on REF under $R^2$. At $\alpha = 10$ the extrapolation ranking collapses onto the global one: the ensemble is EX_EX rank 1 under damage $R^2$ on every tower and rank 1 to 2 under Rel $L^2$ DEL (Table 18). The rank reversal vanishes only when the partition stops measuring extrapolation.

### H.3 The rank reversal survives retraining on two other grids and grows with extrapolation depth

The held-out grid cannot be varied without changing the training set. We therefore retrained the full AutoGluon pool (`best` and `extreme` presets) on two alternative grids for REF, with labels from the unchanged partition algorithm.

- **Grid A (shifted interior indices).** Different interior held-out wind and wave levels over the same envelope; half the training conditions change (144 of the 288). EX_EX holds 240 simulations; merged pool of 93 models.
- **Grid B (wider extrapolation margin).** Two wind levels held out at each envelope edge instead of one, with the paper's wave pattern. EX_EX holds 620 simulations (2.4× the paper's cell); merged pool of 101 models.

The rank reversal survives both grids under both metrics (Table 19). Grid A reproduces the paper pattern: a globally mediocre bagged FastAI network wins EX_EX. On grid B the gap is narrower on the full cell. Splitting by depth explains why: the ensemble keeps rank 1 one step beyond the training

Table 19: **Rank reversal under full retraining on alternative held-out grids** (REF, merged `best+extreme` pool). Grid B is additionally split by extrapolation depth: shallow edge = winds one grid step beyond the training range (288 simulations); deep corner = two steps beyond (318 simulations).The other 14 EX_EX simulations lie at interior wind level 14 but one seed's realized wind statistics fall outside the wind hull; they are excluded from both depth subsets.

| Grid | Quantity | Rel $L^2$ DEL | Damage $R^2$ |
|---|---|---|---|
| A | Global rank-1 (`ExtraTreesMSE_BAG_L1`): EX_EX rank | 20 | 37 |
| | EX_EX rank-1 (`NeuralNetFastAI_r103_BAG_L1`): global rank | 78 | 76 |
| B, full cell | Global rank-1 (`extreme WeightedEnsemble_L2`): EX_EX rank | 6 | 4 |
| | EX_EX rank-1: global rank | 44 (`r191`) | 51 (`r103`) |
| B, shallow edge | Ensemble rank | 1 | 1 |
| B, deep corner | Ensemble rank | 7 | 9 |
| | Deep-corner rank-1 (`NeuralNetFastAI_r191_BAG_L1`) | 0.0744 | 0.932 |

winds but drops to rank 7 to 9 two steps beyond, where a bagged FastAI network is rank 1 under both metrics. The larger cell dilutes the corner. The rank reversal grows with extrapolation depth.

### H.4 Random-split optimism comes from missing extrapolation, not seed leakage

E1 splits by simulation, so all 30 sections of a simulation stay together. Seeds of one wind/wave condition can fall on both sides: $78.6\%$ of E1 test simulations have a same-condition seed in training. This reproduces common practice.

To check whether this seed sharing drives the optimism of E1, we built a condition-grouped random split on REF: $288$ of the $1{,}078$ conditions drawn at random for training with all six seeds kept together (same training size as E1, $1{,}728$ simulations), and retrained the pool. The validation-selected ensemble reaches a global Rel $L^2$ DEL of $0.0198$ ($R^2$ $0.9940$), on par with the simulation-level E1 split ($0.0191$, $R^2$ $0.9944$), and stable across three independent condition draws ($0.0198$–$0.0201$). Both are far below the regime-aware split ($0.063$ global, $0.101$ in EX_EX). The optimism of random splits therefore stems from the absence of extrapolation regimes in the test set, not from seed sharing across the split.

**Summary.** The rank reversal is not an artifact of the partition. Regime labels are stable for $\alpha \in [0.05, 0.3]$, the rank reversal holds in all 15 variants that keep an extrapolation region, and it survives retraining on two other grids, growing with extrapolation depth. At $\alpha = 10$ the ensemble returns to EX_EX rank 1 to 2: the rank reversal vanishes only when the partition stops measuring extrapolation.

# I DO MODELS OUTSIDE AUTOML CHANGE THE PICTURE?

The leaderboards (Appendix G) use AutoGluon as a uniform harness. To rule out a harness artifact, we evaluate classical engineering surrogates (response surface, polynomial chaos, Gaussian process) and standalone modern models (TabPFN, XGBoost with and without monotone constraints) under the identical E2 protocol.

**Protocol.** All baselines use the released E2 split of each tower ($51{,}840$ training rows, same test set and regime labels), the same seven input features (`mean_wind_speed`, `std_wind_speed`, `wave_hs`, `wave_tp`, `section_height_m`, `section_radius_m`, `section_thickness_m`; Appendix E.2) and the same target: models fit DEL $=$ `damage`$^{1/3}$ and damage is the cube of the prediction. Hyperparameters are fixed a priori or chosen on training data only (internal cross-validation or a random $10\%$ hold-out). No test label and no per-regime test score enters fitting or selection.

**Classical surrogates.**

- **Quadratic response surface (RSM)** (Box & Wilson, 1951). Degree-2 polynomial in the seven standardized features (all pairwise interactions), fitted by ridge regression (Hoerl & Kennard, 1970) with the penalty chosen by efficient leave-one-out cross-validation on the training rows over 17 log-spaced values in $[10^{-6}, 10^{2}]$.
- **Polynomial chaos expansion (PCE)** (Xiu & Karniadakis, 2002; Sudret, 2008). Non-intrusive regression PCE with a total-degree Legendre basis (degrees 3 and 4) on inputs mapped to $[-1, 1]$, with ridge-regularized least squares (same penalty grid and selection as RSM).
- **Gaussian process (GP)** (Rasmussen & Williams, 2006). Following GP-based engineering fatigue surrogates (Avendaño-Valencia et al., 2021), one exact GP per tower section (30 GPs per tower), each on the four environmental inputs, with a scaled anisotropic RBF kernel plus a white-noise term and hyperparameters by marginal-likelihood maximization (normalized targets). A single global GP over all seven features was not retained: the length scales collapse under the cross-section dynamic range.

**Standalone modern baselines.**

- **TabPFN** (Hollmann et al., 2025): the pretrained TabPFN-2.5 regressor (Grinsztajn et al., 2025) (`tabpfn` 6.2.0) with its default configuration and pretraining size limits lifted, conditioned on the full $51{,}840$-row training set as context (about 2 minutes per tower on one GPU).
- **XGBoost** (Chen & Guestrin, 2016) (standalone, tuned): up to $3{,}000$ trees, learning rate $0.05$, maximum depth $8$, row and column subsampling $0.8$, early stopping after $100$ rounds on a random $10\%$ hold-out of the training rows.
- **XGBoost with monotone constraints**: the same configuration with monotone-increasing constraints on the turbulence standard deviation (`std_wind_speed`) and the significant wave height (`wave_hs`). This is a diagnostic for tree saturation, not a claim of global monotonicity.

**Results.** Table 20 reports Rel $L^2$ DEL (the primary metric) and Table 21 damage $R^2$, on all test points (Global) and on EX_EX. The two AutoGluon rows are the global rank-1 ensemble (`WeightedEnsemble_L2`) and the EX_EX rank-1 network (`NeuralNetFastAI_r102_BAG_L1`) of Section 5.2.

**Finding 1: no classical surrogate is competitive globally.** Under Rel $L^2$ DEL the classical surrogates reach $0.086$ to $0.147$ globally, against $0.063$ to $0.070$ for the ensemble (damage $R^2$ $0.78$ to $0.90$ against $0.945$ to $0.961$). This supports the focus on modern tabular learners.

**Finding 2: smooth polynomial bases hold up in the extrapolation cell.** On REF, the quadratic response surface beats the ensemble at EX_EX under both metrics ($0.073$ vs $0.101$; $R^2$ $0.967$ vs

Table 20: Additional baselines under E2: Rel $L^2$ DEL (lower is better) on all test points (Global) and on the joint-extrapolation cell (EX_EX). Best value per column in bold.

| | REF | | OPT1 | | OPT2 | |
|---|---|---|---|---|---|---|
| Model | Global | EX_EX | Global | EX_EX | Global | EX_EX |
| *AutoGluon pool (reference)* | | | | | | |
| `WeightedEnsemble_L2` | **0.063** | 0.101 | **0.069** | 0.115 | **0.070** | 0.117 |
| `NeuralNetFastAI_r102_BAG_L1` | 0.079 | **0.054** | 0.085 | **0.075** | 0.081 | **0.073** |
| *Standalone modern* | | | | | | |
| TabPFN | 0.072 | 0.104 | 0.072 | 0.099 | 0.071 | 0.090 |
| XGBoost (tuned) | 0.078 | 0.115 | 0.078 | 0.150 | 0.079 | 0.153 |
| XGBoost + monotone | 0.080 | 0.108 | 0.077 | 0.148 | 0.077 | 0.151 |
| *Classical surrogates* | | | | | | |
| Quadratic RSM | 0.087 | 0.073 | 0.143 | 0.194 | 0.137 | 0.189 |
| PCE (Legendre, degree 3) | 0.087 | 0.074 | 0.108 | 0.117 | 0.104 | 0.118 |
| PCE (Legendre, degree 4) | 0.086 | 0.084 | 0.125 | 0.154 | 0.117 | 0.140 |
| GP (exact, per section) | 0.131 | 0.307 | 0.147 | 0.340 | 0.145 | 0.335 |

Table 21: Additional baselines under E2: damage $R^2$ (higher is better) on all test points (Global) and on the joint-extrapolation cell (EX_EX). Best value per column in bold.

| | REF | | OPT1 | | OPT2 | |
|---|---|---|---|---|---|---|
| Model | Global | EX_EX | Global | EX_EX | Global | EX_EX |
| *AutoGluon pool (reference)* | | | | | | |
| `WeightedEnsemble_L2` | **0.947** | 0.946 | **0.961** | 0.956 | **0.945** | 0.941 |
| `NeuralNetFastAI_r102_BAG_L1` | 0.917 | **0.979** | 0.928 | **0.972** | 0.921 | **0.969** |
| *Standalone modern* | | | | | | |
| TabPFN | 0.922 | 0.913 | 0.953 | 0.946 | 0.940 | 0.956 |
| XGBoost (tuned) | 0.936 | 0.905 | 0.929 | 0.864 | 0.910 | 0.837 |
| XGBoost + monotone | 0.934 | 0.924 | 0.918 | 0.839 | 0.904 | 0.820 |
| *Classical surrogates* | | | | | | |
| Quadratic RSM | 0.904 | 0.967 | 0.836 | 0.874 | 0.821 | 0.879 |
| PCE (Legendre, degree 3) | 0.901 | 0.953 | 0.877 | 0.900 | 0.855 | 0.865 |
| PCE (Legendre, degree 4) | 0.895 | 0.936 | 0.871 | 0.876 | 0.856 | 0.882 |
| GP (exact, per section) | 0.780 | 0.521 | 0.822 | 0.610 | 0.790 | 0.586 |

0.946), and PCE degree 3 is close (0.074). On OPT1 and OPT2 the response surface falls behind, while PCE degree 3 stays on par with the ensemble. The global-versus-extrapolation trade-off thus also appears outside AutoGluon, in mirror image. Classical surrogates are globally crude but hold the trend beyond the envelope. The ensemble is globally sharp but saturates.

**Finding 3: the GP degrades sharply at extrapolation.** The per-section GP is accurate inside the training envelope (Rel $L^2$ DEL 0.072, 0.060 and 0.056 in the In-train cell of REF, OPT1 and OPT2), but its error grows 4 to 6× at EX_EX (0.307 to 0.340; damage $R^2$ 0.52 to 0.61). Its stationary (RBF) kernel reverts toward the prior mean outside the training data.

**Standalone modern baselines.** TabPFN is competitive on OPT1 and OPT2 but does not displace the existing winners: inserted into the released E2 leaderboards (damage $R^2$ ranking), it sits at global ranks 77 / 10 / 5 and EX_EX ranks 79 / 10 / 7 (REF / OPT1 / OPT2). On OPT2 its EX_EX damage $R^2$ (0.956) exceeds the ensemble's (0.941) but remains below the EX_EX rank-1 network (0.969). Standalone XGBoost reproduces the tree pattern of the pool: competitive globally, weaker at EX_EX. Monotone constraints do not recover extrapolation (EX_EX damage $R^2$ changes by ±0.02, Table 24; Rel $L^2$ DEL moves by at most 0.007). Beyond the observed split thresholds, tree predictions stay piecewise constant and bounded by the training leaf values. This points to structural range clipping, not tuning.

**Per-regime breakdown.** Table 22 reports all nine regime cells, scored as in the AutoGluon leaderboards. The ensemble is best in the three wind In-train cells on every tower. The EX_EX rank-1 network is best in all three wind-extrapolation cells on every tower. PCE (degree 3) is best in the

wind-interpolation cells on OPT1 and OPT2. The GP error grows from 0.05–0.08 in the wind In-train cells to 0.17–0.34 once the wind leaves the training range.

Table 22: **Per-regime Rel $L^2$ DEL of the standalone baselines (E2)**, next to the AutoGluon Global rank-1 ensemble and the EX_EX rank-1 network. Columns are wind _ wave regime cells (IT: In-train, IP: Interpolation, EX: Extrapolation); lower is better; bold marks the best model per tower and column.

| Tower | Model | Global | IT_IT | IT_IP | IT_EX | IP_IT | IP_IP | IP_EX | EX_IT | EX_IP | EX_EX |
|---|---|---|---|---|---|---|---|---|---|---|---|
| REF | Ensemble (Global rank-1) | **0.063** | **0.048** | **0.052** | **0.047** | 0.103 | 0.079 | 0.083 | 0.117 | 0.116 | 0.101 |
| | FastAI NN r102 | 0.079 | 0.080 | 0.085 | 0.081 | **0.083** | **0.065** | **0.071** | **0.049** | **0.054** | **0.054** |
| | TabPFN | 0.072 | 0.059 | 0.062 | 0.057 | 0.105 | 0.089 | 0.087 | 0.128 | 0.125 | 0.104 |
| | XGBoost (tuned) | 0.078 | 0.052 | 0.056 | 0.048 | 0.185 | 0.125 | 0.140 | 0.124 | 0.126 | 0.115 |
| | XGBoost (monotone) | 0.080 | 0.049 | 0.054 | 0.047 | 0.197 | 0.148 | 0.158 | 0.118 | 0.119 | 0.108 |
| | PCE (degree 3) | 0.087 | 0.090 | 0.090 | 0.088 | 0.093 | 0.068 | 0.089 | 0.080 | 0.072 | 0.074 |
| | Response surface | 0.087 | 0.089 | 0.091 | 0.089 | 0.085 | 0.071 | 0.072 | 0.075 | 0.078 | 0.073 |
| | GP (per section) | 0.131 | 0.072 | 0.077 | 0.059 | 0.199 | 0.178 | 0.171 | 0.291 | 0.294 | 0.307 |
| OPT1 | Ensemble (Global rank-1) | **0.069** | **0.047** | **0.047** | **0.047** | 0.137 | 0.113 | 0.117 | 0.118 | 0.114 | 0.115 |
| | FastAI NN r102 | 0.085 | 0.082 | 0.079 | 0.080 | 0.137 | 0.113 | 0.120 | **0.080** | **0.081** | **0.075** |
| | TabPFN | 0.072 | 0.056 | 0.055 | 0.055 | 0.142 | 0.123 | 0.123 | 0.102 | 0.100 | 0.099 |
| | XGBoost (tuned) | 0.078 | 0.052 | 0.052 | 0.049 | 0.136 | 0.111 | 0.120 | 0.153 | 0.154 | 0.150 |
| | XGBoost (monotone) | 0.077 | 0.052 | 0.052 | 0.050 | 0.132 | 0.109 | 0.116 | 0.146 | 0.145 | 0.148 |
| | PCE (degree 3) | 0.108 | 0.109 | 0.108 | 0.107 | **0.112** | **0.098** | **0.114** | 0.094 | 0.106 | 0.117 |
| | Response surface | 0.143 | 0.144 | 0.132 | 0.135 | 0.146 | 0.126 | 0.126 | 0.188 | 0.181 | 0.194 |
| | GP (per section) | 0.147 | 0.060 | 0.062 | 0.052 | 0.271 | 0.254 | 0.236 | 0.320 | 0.321 | 0.340 |
| OPT2 | Ensemble (Global rank-1) | **0.070** | **0.047** | **0.049** | **0.049** | 0.139 | 0.116 | 0.119 | 0.117 | 0.115 | 0.117 |
| | FastAI NN r102 | 0.081 | 0.076 | 0.075 | 0.075 | 0.136 | 0.113 | 0.119 | **0.080** | **0.082** | **0.073** |
| | TabPFN | 0.071 | 0.057 | 0.056 | 0.055 | 0.137 | 0.117 | 0.118 | 0.102 | 0.100 | 0.090 |
| | XGBoost (tuned) | 0.079 | 0.051 | 0.053 | 0.050 | 0.135 | 0.112 | 0.118 | 0.154 | 0.156 | 0.153 |
| | XGBoost (monotone) | 0.077 | 0.050 | 0.052 | 0.049 | 0.128 | 0.108 | 0.115 | 0.147 | 0.148 | 0.151 |
| | PCE (degree 3) | 0.104 | 0.104 | 0.104 | 0.103 | **0.107** | **0.095** | **0.111** | 0.092 | 0.105 | 0.118 |
| | Response surface | 0.137 | 0.139 | 0.127 | 0.131 | 0.135 | 0.115 | 0.117 | 0.179 | 0.173 | 0.189 |
| | GP (per section) | 0.145 | 0.056 | 0.060 | 0.051 | 0.278 | 0.247 | 0.235 | 0.315 | 0.316 | 0.335 |

**Summary.** None of the added baselines changes the E2 conclusions: the global and EX_EX rank-1 remain different models on every tower.

# J CAN VALIDATION PICK THE EXTRAPOLATION WINNER, AND WHY DO ENSEMBLES FAIL?

This appendix quantifies two aspects of the E2 rank reversal (Section 5.2): what in-distribution model selection costs at the joint-extrapolation cell (EX_EX), and why the validation-selected ensemble degrades there. Both analyses reuse the stored per-row test predictions of the released pools (95/96/95 models on REF/OPT1/OPT2); no model is retrained.

## J.1 VALIDATION SELECTION LEAVES 1.5 TO 1.9 TIMES THE BEST EXTRAPOLATION ERROR

**Selection is fit in-distribution by design.** In all FLOATBench protocols, the ensemble weights and the deployed model are chosen on validation data from the training distribution. This is deliberate: a practitioner has no labeled extrapolation data. The resulting objective mismatch is part of the measurement, not a confound, and the regime-aware protocol makes it reproducible.

**Selection regret.** For a selection rule $s$, we define the regret at the joint-extrapolation cell as

$$\text{regret}(s) \;=\; \max_{m} R^2_{\text{EX_EX}}(m) \;-\; R^2_{\text{EX_EX}}\big(m_s\big),$$

where $m_s$ is the model picked by $s$ and the maximum runs over the pool. We compare three rules: (i) the paper protocol (validation score, which selects `WeightedEnsemble_L2` on every tower); (ii) the best model on the In-train test cell (IT_IT); and (iii) the best model on the joint-interpolation test cell (IP_IP). Rules (ii) and (iii) read test labels of those cells, which a practitioner would not have either; they are diagnostic upper bounds on "nearest-available-distribution" selection, not deployable methods.

Table 23: Selection regret at EX_EX (damage $R^2$). EX rank is the rank of the selected model within the EX_EX leaderboard of the pool. The oracle is `NeuralNetFastAI_r102_BAG_L1` on all three towers ($R^2_{\text{EX_EX}} = 0.979/0.972/0.969$).

| Tower | Selection rule | EX rank | EX $R^2$ | Regret |
|---|---|---|---|---|
| REF | Validation (paper protocol) | 10 | 0.946 | 0.033 |
| | Best on IT_IT | 85 | 0.890 | 0.089 |
| | Best on IP_IP | 2 | 0.979 | 0.000 |
| OPT1 | Validation (paper protocol) | 7 | 0.956 | 0.015 |
| | Best on IT_IT | 88 | 0.635 | 0.337 |
| | Best on IP_IP | 65 | 0.895 | 0.077 |
| OPT2 | Validation (paper protocol) | 8 | 0.941 | 0.027 |
| | Best on IT_IT | 95 | −0.366 | 1.334 |
| | Best on IP_IP | 72 | 0.856 | 0.112 |

Validation selection leaves 0.015 to 0.033 of EX_EX $R^2$ on every tower (Table 23). Selecting on IT_IT is catastrophic. That cell is near-saturated, so its ranking says little about extrapolation. Selecting on IP_IP is inconsistent: near-exact on REF (regret $1 \times 10^{-4}$), worse than validation on OPT1/OPT2. Kendall's $\tau$ between either proxy ranking and the EX_EX ranking lies in $[-0.14, 0.25]$. Performance inside or between training conditions does not predict performance beyond them.

**Under the primary metric.** The picture is unchanged under Rel $L^2$ DEL. The ensemble reaches an EX_EX error of 0.1015/0.1149/0.1170 (rank 23/11/11), against 0.0543/0.0752/0.0730 for the best model: 1.87×/1.53×/1.60× the best available error. The IT_IT rule picks EX_EX ranks 44/89/90, the IP_IP rule ranks 4/8/8, and every proxy-versus-EX_EX $\tau$ is at most 0.34. No tested rule reliably finds the good extrapolator. OOD-aware model selection for engineering surrogates is an open problem that FLOATBench makes measurable.

## J.2 THE ENSEMBLE LEANS ON TREES, AND TREES UNDER-PREDICT BEYOND THE TRAINING DOMAIN

We decompose the failure into three measured steps (Table 24).

**Steps 1 and 2: the ensemble sits on trees, and trees under-predict.** The validation-fitted ensemble puts 86 to 88% of its weight on tree models, the strongest in-distribution; the rest goes to neural networks. At EX_EX every family under-predicts on median (prediction minus truth, normalized target units), but trees more so on every tower. Under-prediction is the non-conservative direction for fatigue design. Monotone constraints on XGBoost (`std_wind_speed`, `wave_hs`) do not fix this and leave it far from the oracle (Appendix I): the failure is range clipping, not mis-ordering.

**Step 3: the best extrapolators are smooth models.** BAG_L1 neural networks (MLP and `TabM` backbones) are about one fifth of each pool but take 8/6/8 of the EX_EX top-10 slots by damage $R^2$. Under Rel $L^2$ DEL all ten EX_EX top slots are neural networks (Appendix K.2). `NeuralNetFastAI_r102_BAG_L1` is EX_EX rank-1 on all towers at global ranks 79/73/69 under Rel $L^2$ DEL, so no evaluated in-distribution criterion would pick it. The response surface and polynomial chaos also degrade far less from global to extrapolation than the ensemble and the trees (Appendix I).

**Caveat: the neural-network family is high-variance.** The claim is not that neural networks extrapolate better as a family. The worst extrapolators are also neural networks (last EX_EX rank on every tower; on OPT2 it is `TabM_BAG_L1`, $R^2_{\text{EX_EX}} = -0.37$), and on OPT1 their median rank is worse than that of trees. Trees cluster at mid ranks with a consistent negative bias. Neural networks have heavy tails in both directions. Validation cannot find the good tail; the regime labels can.

Table 24: Mechanism diagnostics at EX_EX. Tree weight: share of `WeightedEnsemble_L2` weight on tree-based base models. Bias: median signed error (normalized target units). Neural networks: L1 base learners (`NeuralNetFastAI`, `NeuralNetTorch` and `TabM` BAG_L1; 14/13/13 MLPs plus 6 `TabM` per pool); BAG_L2 stackers are excluded, so the pool counts are below the NN-F plus NN-T totals of Table 10, which include them. NN top-10: number of these networks among the ten best EX_EX models of the full pool, ranked by damage $R^2$. Monotone $\Delta R^2$: EX_EX damage $R^2$ of monotone-constrained minus unconstrained XGBoost.

| | REF | OPT1 | OPT2 |
|---|---|---|---|
| Ensemble weight on trees | 87.5% | 87.5% | 85.7% |
| Median bias, trees | −0.061 | −0.164 | −0.202 |
| Median bias, neural networks | −0.009 | −0.117 | −0.108 |
| Neural networks in pool | 20/95 | 19/96 | 19/95 |
| Neural networks in EX_EX top-10 | 8 | 6 | 8 |
| Median EX_EX rank, NN / trees | 26.5 / 38.0 | 65.0 / 61.5 | 15.0 / 64.0 |
| Monotone XGBoost $\Delta R^2$ | +0.02 | −0.02 | −0.02 |

**Summary.** No tested selection rule reliably picks the extrapolation winner: the validation-selected ensemble reaches 1.53 to 1.87× the best EX_EX error. The ensemble fails because it puts 86 to 88% of its weight on trees, which under-predict beyond the training domain. The EX_EX rank-1 is a bagged FastAI network that no evaluated in-distribution criterion would pick, but the worst extrapolators are also neural networks.

# K WHICH SURROGATE WINS DEPENDS ON THE REGIME AND THE SECTION (E2)

## K.1 JOINT WIND AND WAVE EXTRAPOLATION HAS THE HIGHEST MRE ON EVERY TOWER

The extreme wind and wave regime (EX_EX) is the deepest extrapolation cell by construction. Figure 12 shows the top-10 global surrogates (sorted by Rel $L^2$ DEL, Appendix G.2) and their MRE DEL across the nine regimes per tower; the rightmost EX_EX column has the highest MRE on all three towers. This holds for MRE. Under Rel $L^2$ the three wind-extrapolation cells are of similar difficulty and wave extrapolation alone stays near In-train error (Table 22), so the wind axis carries the extrapolation difficulty.

## K.2 EVERY EX_EX TOP-10 MODEL IS A BAGGED NEURAL NETWORK

Table 25 lists the top-10 surrogates ranked by Rel $L^2$ DEL on the EX_EX regime per tower. Every top-10 EX_EX model is a `NeuralNetFastAI` or `NeuralNetTorch` BAG_L1 variant; most have low global ranks (bottom of the global leaderboard), while OPT2 includes two higher-global-rank `NeuralNetTorch` variants. The family-level signature is examined in Appendix K.3.

Table 25: **Top-10 EX_EX (E2) per tower, sorted by Rel $L^2$ DEL on EX_EX.** Error metric on the DEL target.

| Rank EX_EX | Rank Global | Model | Preset | Rel $L^2$ Global | Rel $L^2$ EX_EX |
|---|---|---|---|---|---|
| REF | | | | | |
| 1 | 79 | NeuralNetFastAI_r102_BAG_L1 | best | 0.079 | **0.054** |
| 2 | 84 | NeuralNetFastAI_r156_BAG_L1 | best | 0.082 | **0.062** |
| 3 | 80 | NeuralNetFastAI_r11_BAG_L1 | best | 0.079 | **0.063** |
| 4 | 77 | NeuralNetFastAI_r103_BAG_L1 | best | 0.077 | **0.064** |
| 5 | 83 | NeuralNetTorch_r30_BAG_L1 | best | 0.081 | **0.065** |
| 6 | 74 | NeuralNetFastAI_r145_BAG_L1 | best | 0.076 | **0.077** |
| 7 | 82 | NeuralNetTorch_r86_BAG_L1 | best | 0.081 | **0.077** |
| 8 | 73 | NeuralNetFastAI_r191_BAG_L1 | best | 0.075 | **0.080** |
| 9 | 76 | NeuralNetFastAI_BAG_L1 | best | 0.077 | **0.084** |
| 10 | 68 | NeuralNetTorch_r79_BAG_L1 | best | 0.072 | **0.089** |
| OPT1 | | | | | |
| 1 | 73 | NeuralNetFastAI_r102_BAG_L1 | best | 0.085 | **0.075** |
| 2 | 80 | NeuralNetFastAI_r191_BAG_L1 | best | 0.089 | **0.085** |
| 3 | 77 | NeuralNetFastAI_BAG_L1 | best | 0.087 | **0.086** |
| 4 | 78 | NeuralNetFastAI_r145_BAG_L1 | best | 0.088 | **0.086** |
| 5 | 76 | NeuralNetFastAI_r103_BAG_L1 | best | 0.087 | **0.090** |
| 6 | 85 | NeuralNetTorch_r30_BAG_L1 | best | 0.107 | **0.102** |
| 7 | 64 | NeuralNetTorch_BAG_L1 | best | 0.078 | **0.106** |
| 8 | 82 | NeuralNetFastAI_r11_BAG_L1 | best | 0.092 | **0.106** |
| 9 | 79 | NeuralNetTorch_r14_BAG_L1 | best | 0.089 | **0.110** |
| 10 | 71 | NeuralNetTorch_r22_BAG_L1 | best | 0.083 | **0.113** |
| OPT2 | | | | | |
| 1 | 69 | NeuralNetFastAI_r102_BAG_L1 | best | 0.081 | **0.073** |
| 2 | 79 | NeuralNetFastAI_r191_BAG_L1 | best | 0.089 | **0.079** |
| 3 | 71 | NeuralNetFastAI_BAG_L1 | best | 0.085 | **0.081** |
| 4 | 80 | NeuralNetFastAI_r103_BAG_L1 | best | 0.089 | **0.081** |
| 5 | 75 | NeuralNetFastAI_r145_BAG_L1 | best | 0.088 | **0.086** |
| 6 | 83 | NeuralNetTorch_r30_BAG_L1 | best | 0.103 | **0.094** |
| 7 | 78 | NeuralNetFastAI_r11_BAG_L1 | best | 0.089 | **0.096** |
| 8 | 84 | NeuralNetFastAI_r143_BAG_L1 | best | 0.104 | **0.099** |
| 9 | 8 | NeuralNetTorch_BAG_L1 | best | 0.074 | **0.103** |
| 10 | 44 | NeuralNetTorch_r22_BAG_L1 | best | 0.077 | **0.110** |

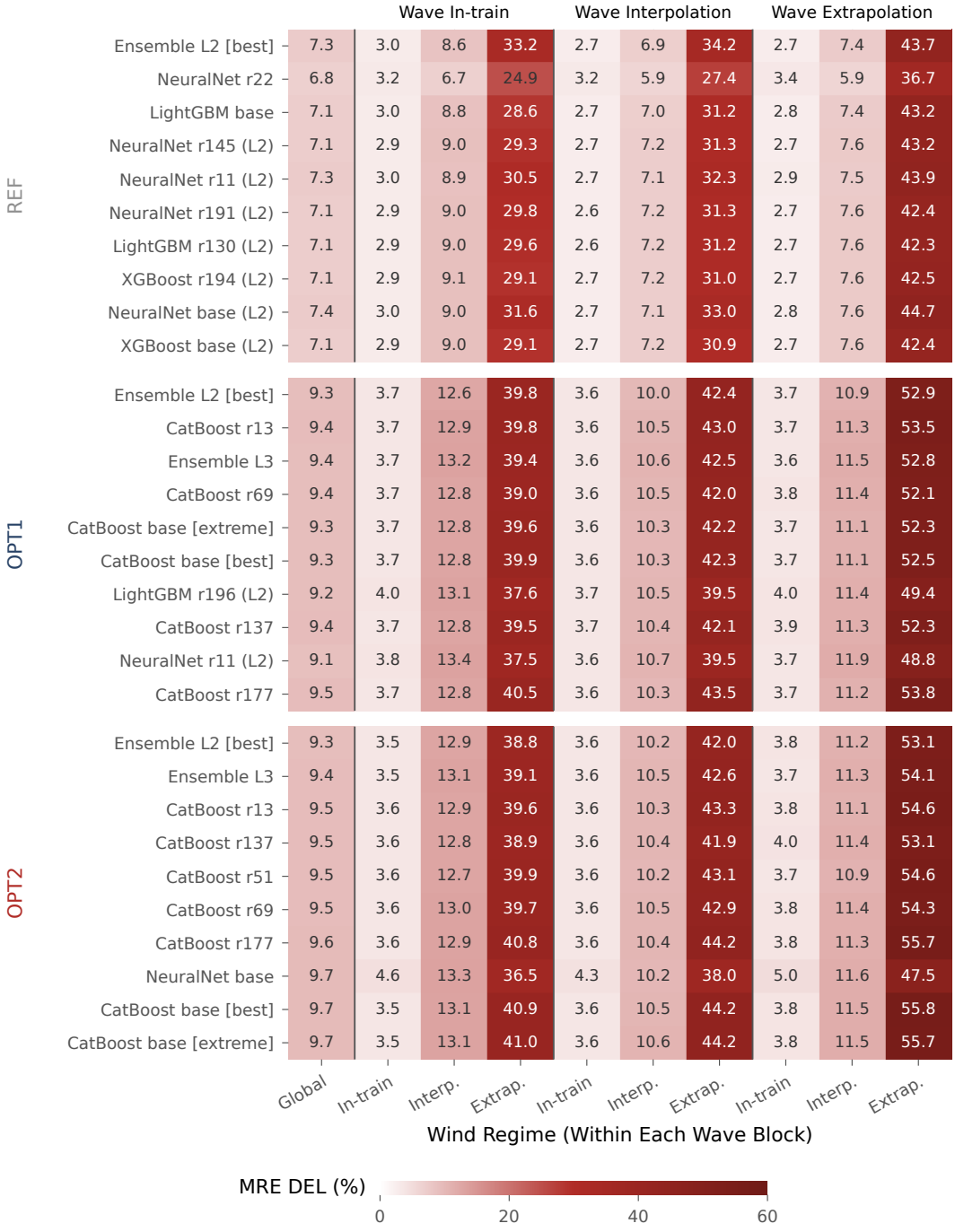



Figure 12: **MRE DEL across the nine regimes, three towers (E2).** Top-10 global surrogates per tower (sorted by Rel $L^2$ DEL, Appendix G.2). Panels: REF, OPT1, OPT2 (top to bottom). Columns group wave regimes; within each group, the three sub-columns are wind regimes. The rightmost EX_EX column is the deepest extrapolation cell by construction.

### K.3 NEURAL NETWORKS DEGRADE LEAST FROM GLOBAL TO EX_EX; TABM DEGRADES MOST

Figure 13 shows the per-model MRE DEL on Global (y-axis) vs. EX_EX (x-axis) for the three towers. Each point is one of the ∼95 surrogates; color marks the family. Points below the diagonal have higher EX_EX error than Global error; models closer to the diagonal degrade less. `NeuralNet` variants stay closest to the diagonal, while `TabM` departs farthest below, consistent with the family-level pattern reported in the main text.

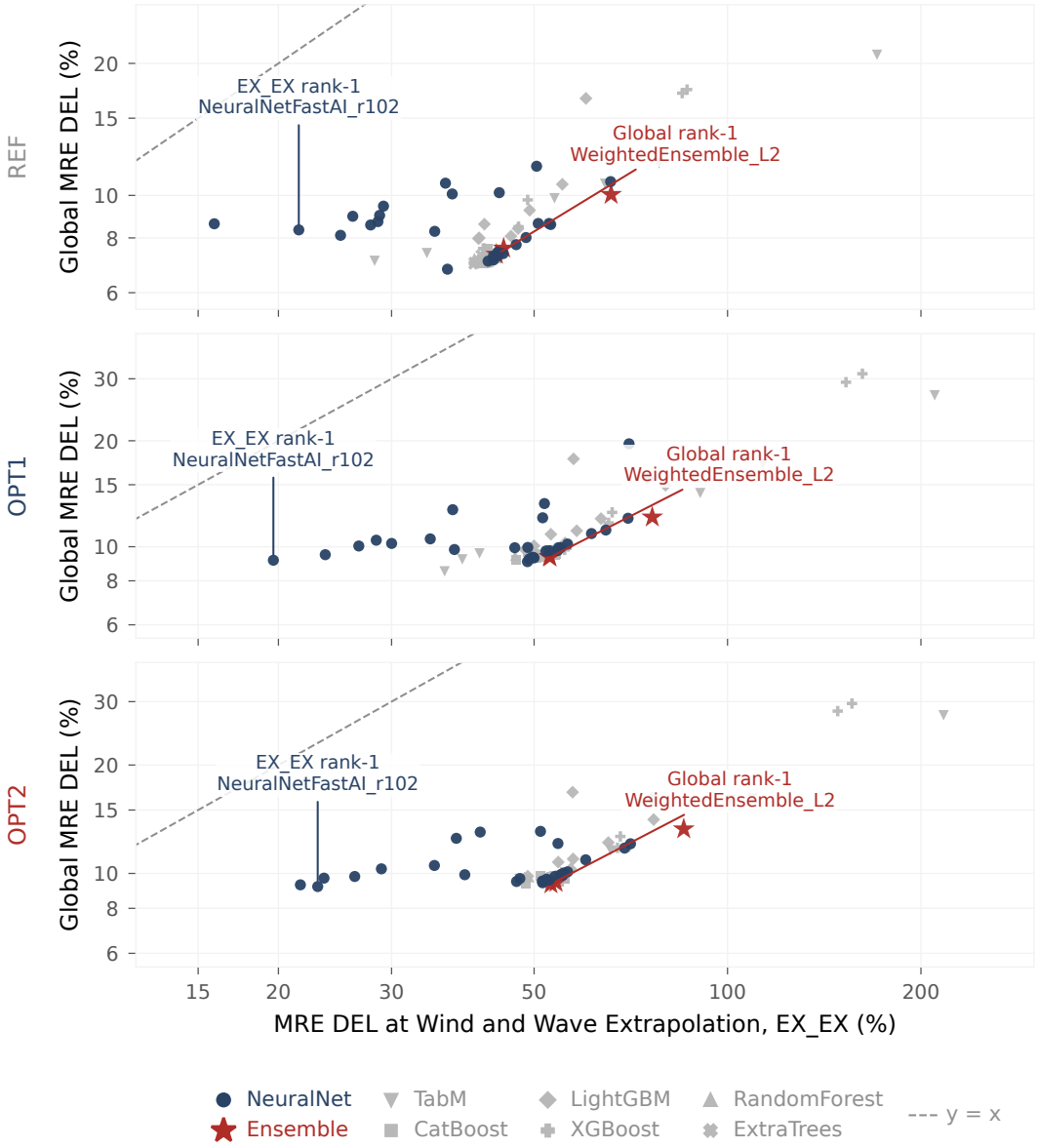



Figure 13: **Per-model MRE DEL: Global (y) vs. EX_EX (x), three towers (E2).** All ∼95 surrogates per tower, marker by family (`NeuralNet` navy, ensembles red, other families grey); panels REF, OPT1, OPT2 top to bottom, log axes. Points below the diagonal have higher EX_EX error than Global error; models closer to the diagonal degrade less. `NeuralNet` variants stay closest to the diagonal, while `TabM` departs farthest below.

### K.4 WIND EXTRAPOLATION DRIVES THE ERROR; THE WAVE AXIS STAYS FLAT

Figure 14 shows the family-aggregated MRE DEL split by wind regime (left column) and wave regime (right column) for the three towers; `NeuralNet` attains the lowest wind EX MRE DEL and `TabM` the highest on every tower, confirming the per-surrogate signature in Appendix K.3.

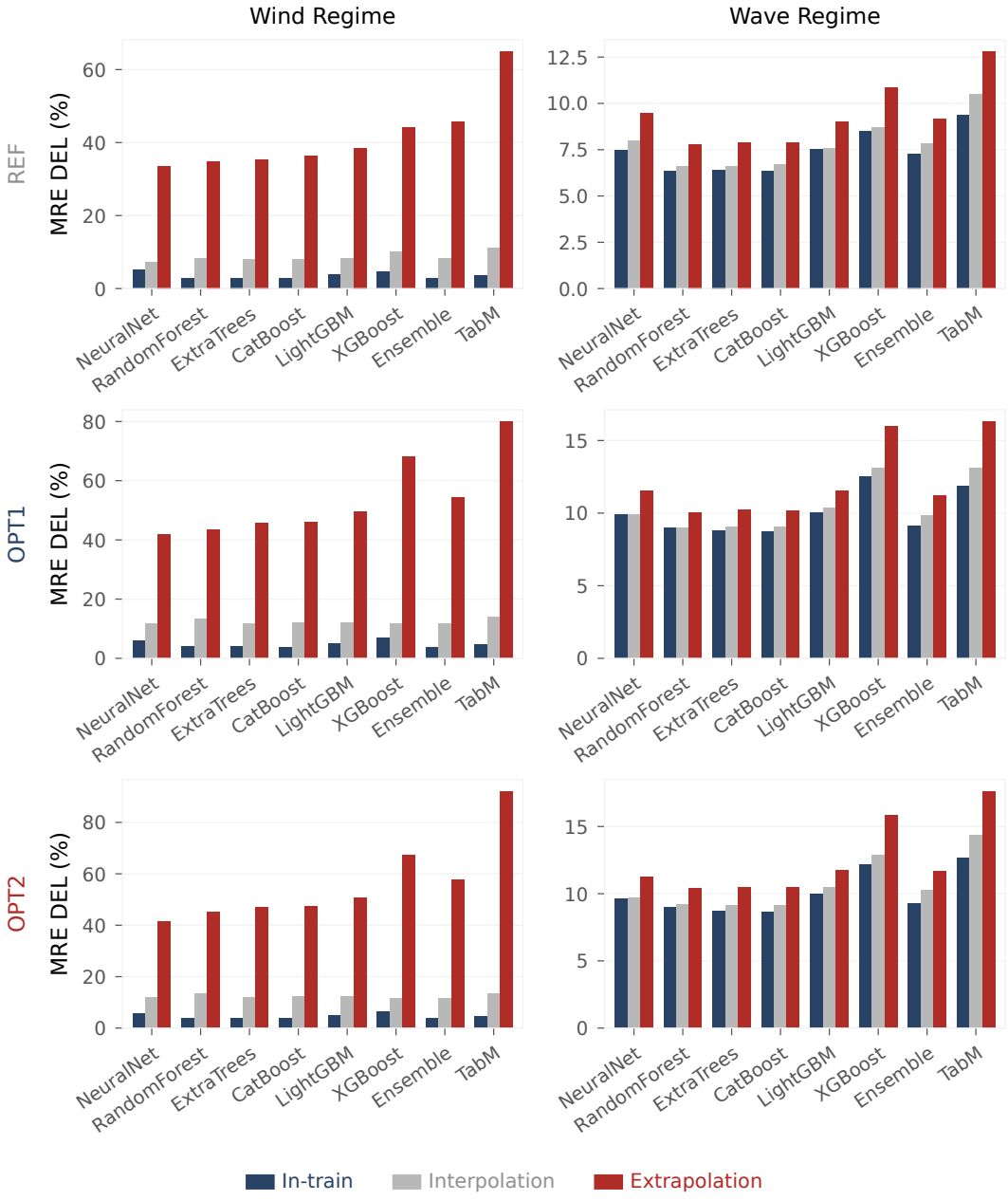



Figure 14: **Family-aggregated MRE DEL by regime, three towers (E2).** ROWS: REF, OPT1, OPT2; left column: wind regime; right column: wave regime. The wind axis drives the bulk of the family-level extrapolation error; the wave axis is comparatively flat.

### K.5 THE BASE AND THE TOP PICK DIFFERENT WINNERS

Table 26 and Table 27 report the top-10 surrogates ranked by Section 1 (base) and Section 30 (top) Rel $L^2$ DEL respectively. At the base, `NeuralNetFastAI` BAG_L2 and `NeuralNetTorch` variants dominate on REF, while `WeightedEnsemble_L2` retains rank 1 on OPT1/OPT2. At the top, `RandomForest` variants dominate REF, and `NeuralNetTorch_r22_BAG_L1` consistently wins on OPT1/OPT2.

Table 26: **Top-10 Section 1 (E2) per tower, sorted by Rel $L^2$ DEL on Section 1.** Error metric on the DEL target.

| Rank Sec. 1 | Rank Global | Model | Preset | Rel $L^2$ Global | Rel $L^2$ Sec. 1 |
|---|---|---|---|---|---|
| REF | | | | | |
| 1 | 5 | NeuralNetFastAI_r11_BAG_L2 | best | 0.0638 | **0.0601** |
| 2 | 2 | NeuralNetTorch_r22_BAG_L1 | best | 0.0635 | **0.0601** |
| 3 | 4 | NeuralNetFastAI_r145_BAG_L2 | best | 0.0637 | **0.0602** |
| 4 | 3 | LightGBMLarge_BAG_L1 | best | 0.0636 | **0.0603** |
| 5 | 16 | CatBoost_r13_BAG_L2 | best | 0.0641 | **0.0603** |
| 6 | 11 | CatBoost_r177_BAG_L2 | best | 0.0641 | **0.0604** |
| 7 | 1 | WeightedEnsemble_L2 | best | 0.0634 | **0.0604** |
| 8 | 6 | NeuralNetFastAI_r191_BAG_L2 | best | 0.0639 | **0.0605** |
| 9 | 20 | RandomForest_r195_BAG_L1 | best | 0.0642 | **0.0605** |
| 10 | 27 | NeuralNetTorch_r79_BAG_L2 | best | 0.0643 | **0.0606** |
| OPT1 | | | | | |
| 1 | 1 | WeightedEnsemble_L2 | best | 0.0692 | **0.0761** |
| 2 | 7 | LightGBM_r196_BAG_L2 | best | 0.0719 | **0.0764** |
| 3 | 4 | CatBoost_r69_BAG_L1 | best | 0.0716 | **0.0775** |
| 4 | 8 | CatBoost_r137_BAG_L1 | best | 0.0721 | **0.0776** |
| 5 | 5 | CatBoost_BAG_L1 | extreme | 0.0716 | **0.0778** |
| 6 | 2 | CatBoost_r13_BAG_L1 | best | 0.0712 | **0.0778** |
| 7 | 6 | CatBoost_BAG_L1 | best | 0.0717 | **0.0778** |
| 8 | 3 | WeightedEnsemble_L3 | best | 0.0712 | **0.0780** |
| 9 | 10 | CatBoost_r177_BAG_L1 | best | 0.0724 | **0.0785** |
| 10 | 29 | XGBoost_BAG_L1 | best | 0.0744 | **0.0788** |
| OPT2 | | | | | |
| 1 | 1 | WeightedEnsemble_L2 | best | 0.0704 | **0.0785** |
| 2 | 4 | CatBoost_r137_BAG_L1 | best | 0.0727 | **0.0793** |
| 3 | 20 | XGBoost_r194_BAG_L1 | best | 0.0757 | **0.0796** |
| 4 | 3 | CatBoost_r13_BAG_L1 | best | 0.0719 | **0.0798** |
| 5 | 2 | WeightedEnsemble_L3 | best | 0.0719 | **0.0798** |
| 6 | 6 | CatBoost_r69_BAG_L1 | best | 0.0730 | **0.0802** |
| 7 | 5 | CatBoost_r51_BAG_L1 | extreme | 0.0727 | **0.0803** |
| 8 | 7 | CatBoost_r177_BAG_L1 | best | 0.0735 | **0.0811** |
| 9 | 30 | XGBoost_BAG_L1 | best | 0.0763 | **0.0811** |
| 10 | 9 | CatBoost_BAG_L1 | best | 0.0738 | **0.0812** |

**Summary.** Which surrogate wins depends on the regime and the section. EX_EX, the deepest extrapolation cell, has the highest MRE on every tower, and every EX_EX top-10 model is a bagged neural network, most of them low on the global leaderboard. Wind extrapolation drives the error, and the base and the top pick different winners.

Table 27: **Top-10 Section 30 (E2) per tower, sorted by Rel L$^2$ DEL on Section 30.** Error metric on the DEL target.

| Rank Sec. 30 | Rank Global | Model | Preset | Rel L$^2$ Global | Rel L$^2$ Sec. 30 |
|---|---|---|---|---|---|
| REF | | | | | |
| 1 | 60 | RandomForestMSE_BAG_L1 | best | 0.0670 | **0.0456** |
| 2 | 20 | RandomForest_r195_BAG_L1 | best | 0.0642 | **0.0462** |
| 3 | 34 | RandomForestMSE_BAG_L2 | best | 0.0646 | **0.0466** |
| 4 | 26 | RandomForest_r195_BAG_L2 | best | 0.0643 | **0.0468** |
| 5 | 19 | LightGBM_BAG_L2 | best | 0.0642 | **0.0471** |
| 6 | 18 | LightGBM_r131_BAG_L2 | best | 0.0642 | **0.0473** |
| 7 | 7 | LightGBM_r130_BAG_L2 | best | 0.0640 | **0.0474** |
| 8 | 11 | CatBoost_r177_BAG_L2 | best | 0.0641 | **0.0476** |
| 9 | 15 | LightGBMLarge_BAG_L2 | best | 0.0641 | **0.0477** |
| 10 | 17 | LightGBM_r161_BAG_L2 | best | 0.0642 | **0.0477** |
| OPT1 | | | | | |
| 1 | 71 | NeuralNetTorch_r22_BAG_L1 | best | 0.0832 | **0.0523** |
| 2 | 1 | WeightedEnsemble_L2 | best | 0.0692 | **0.0531** |
| 3 | 3 | WeightedEnsemble_L3 | best | 0.0712 | **0.0554** |
| 4 | 2 | CatBoost_r13_BAG_L1 | best | 0.0712 | **0.0559** |
| 5 | 33 | LightGBM_r130_BAG_L2 | best | 0.0749 | **0.0571** |
| 6 | 5 | CatBoost_BAG_L1 | extreme | 0.0716 | **0.0574** |
| 7 | 6 | CatBoost_BAG_L1 | best | 0.0717 | **0.0575** |
| 8 | 48 | RandomForest_r195_BAG_L1 | best | 0.0757 | **0.0577** |
| 9 | 4 | CatBoost_r69_BAG_L1 | best | 0.0716 | **0.0578** |
| 10 | 10 | CatBoost_r177_BAG_L1 | best | 0.0724 | **0.0580** |
| OPT2 | | | | | |
| 1 | 44 | NeuralNetTorch_r22_BAG_L1 | best | 0.0769 | **0.0547** |
| 2 | 1 | WeightedEnsemble_L2 | best | 0.0704 | **0.0580** |
| 3 | 12 | LightGBM_r96_BAG_L2 | best | 0.0750 | **0.0593** |
| 4 | 2 | WeightedEnsemble_L3 | best | 0.0719 | **0.0597** |
| 5 | 34 | LightGBM_BAG_L2 | best | 0.0765 | **0.0603** |
| 6 | 3 | CatBoost_r13_BAG_L1 | best | 0.0719 | **0.0606** |
| 7 | 8 | NeuralNetTorch_BAG_L1 | best | 0.0737 | **0.0607** |
| 8 | 39 | LightGBM_r130_BAG_L2 | best | 0.0767 | **0.0614** |
| 9 | 17 | LightGBM_r188_BAG_L2 | best | 0.0755 | **0.0615** |
| 10 | 7 | CatBoost_r177_BAG_L1 | best | 0.0735 | **0.0615** |

# L DOES THE WINNER SURVIVE A CHANGE OF TOWER? (E3)

## L.1 TRANSFER HOLDS WHEN REF IS IN TRAINING AND UNDER-PREDICTS WHEN IT IS HELD OUT

Figure 15 shows the predicted-vs-true damage scatter for the top-3 Global surrogates per fold (sorted by Rel $L^2$ DEL). The folds that include REF in training (top two rows) sit on the diagonal; the fold that holds REF out (bottom row) under-predicts, consistent with REF's wider damage profile and most-distinct geometry (Figure 4).

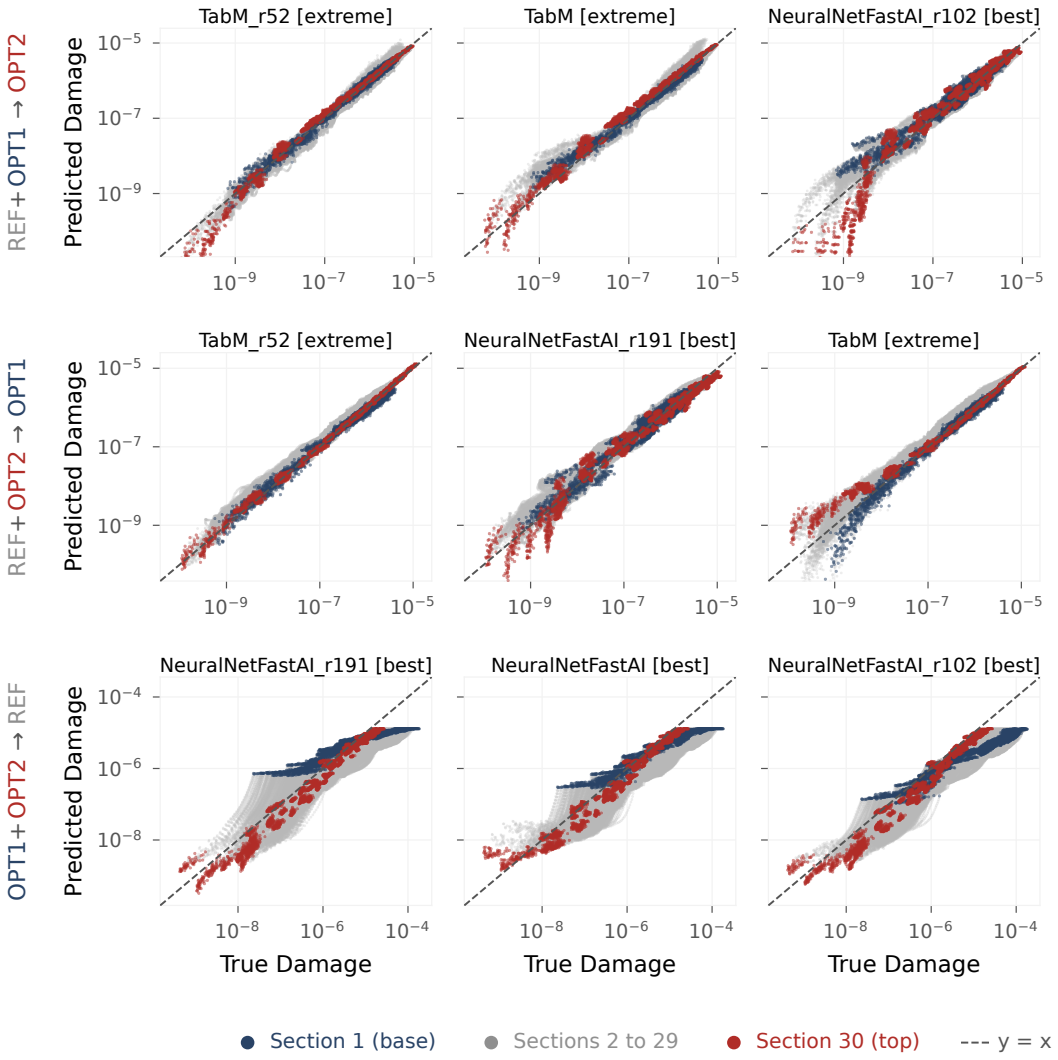



Figure 15: **Predicted-vs-true damage scatter for the top-3 Global surrogates per cross-tower fold (E3), sorted by Rel $L^2$ DEL.** Rows: folds; columns: top-3 Global surrogates; log-log axes. The bottom fold under-predicts the held-out tower; the top two folds are close to the diagonal.

### L.2 TabM wins the top on every fold; neural networks win the base

Table 28 and Table 29 report the top-10 cross-tower surrogates ranked by Section 1 (base) and Section 30 (top) Rel $L^2$ DEL respectively. `TabM_r52_BAG_L1` (extreme) dominates Section 30 on all three folds; the Section 1 winners are `NeuralNet` variants on all folds: stacked BAG_L2 variants on the easier folds (with `LightGBM` variants close behind), and a BAG_L1 `NeuralNet` variant on OPT1+OPT2 → REF.

Table 28: **Top-10 Section 1 (E3) per fold, sorted by Rel $L^2$ DEL on Section 1.** Error metric on the DEL target.

| Rank Sec. 1 | Rank Global | Model | Preset | Rel $L^2$ Global | Rel $L^2$ Sec. 1 |
|---|---|---|---|---|---|
| REF + OPT1 → OPT2 | | | | | |
| 1 | 56 | NeuralNetTorch_r22_BAG_L2 | best | 0.6167 | **0.0298** |
| 2 | 43 | NeuralNetFastAI_r102_BAG_L2 | best | 0.4541 | **0.0302** |
| 3 | 31 | LightGBM_BAG_L1 | best | 0.4318 | **0.0303** |
| 4 | 32 | XGBoost_r33_BAG_L2 | best | 0.4357 | **0.0307** |
| 5 | 37 | XGBoost_BAG_L2 | best | 0.4422 | **0.0309** |
| 6 | 28 | CatBoost_BAG_L2 | best | 0.3976 | **0.0310** |
| 7 | 41 | LightGBMLarge_BAG_L1 | best | 0.4461 | **0.0310** |
| 8 | 26 | LightGBMXT_BAG_L2 | best | 0.3914 | **0.0311** |
| 9 | 25 | CatBoost_r177_BAG_L2 | best | 0.3842 | **0.0311** |
| 10 | 33 | LightGBM_r131_BAG_L2 | best | 0.4364 | **0.0312** |
| REF + OPT2 → OPT1 | | | | | |
| 1 | 42 | NeuralNetFastAI_BAG_L2 | best | 0.2251 | **0.0305** |
| 2 | 47 | LightGBMLarge_BAG_L1 | best | 0.2403 | **0.0306** |
| 3 | 38 | LightGBMLarge_BAG_L2 | best | 0.2132 | **0.0309** |
| 4 | 41 | LightGBM_BAG_L2 | best | 0.2219 | **0.0310** |
| 5 | 36 | LightGBM_r131_BAG_L2 | best | 0.2070 | **0.0310** |
| 6 | 32 | CatBoost_r177_BAG_L2 | best | 0.1967 | **0.0310** |
| 7 | 26 | ExtraTrees_r42_BAG_L2 | best | 0.1741 | **0.0311** |
| 8 | 31 | CatBoost_BAG_L2 | best | 0.1966 | **0.0311** |
| 9 | 27 | ExtraTreesMSE_BAG_L2 | best | 0.1816 | **0.0311** |
| 10 | 15 | LightGBM_BAG_L1 | best | 0.1355 | **0.0311** |
| OPT1 + OPT2 → REF | | | | | |
| 1 | 2 | NeuralNetFastAI_BAG_L1 | best | 0.4332 | **0.4288** |
| 2 | 1 | NeuralNetFastAI_r191_BAG_L1 | best | 0.4228 | **0.4311** |
| 3 | 3 | NeuralNetFastAI_r102_BAG_L1 | best | 0.4903 | **0.4915** |
| 4 | 4 | NeuralNetTorch_BAG_L1 | best | 0.5384 | **0.5631** |
| 5 | 6 | NeuralNetFastAI_r145_BAG_L1 | best | 0.5604 | **0.5872** |
| 6 | 7 | NeuralNetTorch_r22_BAG_L1 | best | 0.5687 | **0.5892** |
| 7 | 5 | NeuralNetTorch_r79_BAG_L1 | best | 0.5585 | **0.5980** |
| 8 | 8 | LightGBMXT_BAG_L1 | best | 0.5927 | **0.6510** |
| 9 | 12 | WeightedEnsemble_L3 | best | 0.6026 | **0.6510** |
| 10 | 13 | WeightedEnsemble_L2 | best | 0.6026 | **0.6510** |

**Summary.** The winner survives a change of tower only in part. Transfer holds when REF is in training, but the fold that holds REF out under-predicts. By section, `TabM_r52_BAG_L1` wins the top on every fold, while `NeuralNet` variants win the base.

Table 29: **Top-10 Section 30 (E3) per fold, sorted by Rel $L^2$ DEL on Section 30.** Error metric on the DEL target.

| Rank Sec. 30 | Rank Global | Model | Preset | Rel $L^2$ Global | Rel $L^2$ Sec. 30 |
|---|---|---|---|---|---|
| REF + OPT1 → OPT2 | | | | | |
| 1 | 1 | TabM_r52_BAG_L1 | extreme | 0.0670 | **0.0323** |
| 2 | 13 | WeightedEnsemble_L2 | extreme | 0.1142 | **0.0469** |
| 3 | 2 | TabM_BAG_L1 | extreme | 0.0808 | **0.0573** |
| 4 | 31 | LightGBM_BAG_L1 | best | 0.4318 | **0.0623** |
| 5 | 44 | CatBoost_r9_BAG_L1 | best | 0.4729 | **0.0627** |
| 6 | 7 | NeuralNetTorch_r22_BAG_L1 | best | 0.0989 | **0.0676** |
| 7 | 18 | LightGBM_r188_BAG_L1 | best | 0.1757 | **0.0753** |
| 8 | 17 | LightGBMXT_BAG_L1 | best | 0.1632 | **0.0780** |
| 9 | 3 | NeuralNetFastAI_r102_BAG_L1 | best | 0.0870 | **0.0853** |
| 10 | 8 | ExtraTreesMSE_BAG_L1 | best | 0.0993 | **0.0864** |
| REF + OPT2 → OPT1 | | | | | |
| 1 | 1 | TabM_r52_BAG_L1 | extreme | 0.0975 | **0.0259** |
| 2 | 3 | TabM_BAG_L1 | extreme | 0.1094 | **0.0344** |
| 3 | 13 | WeightedEnsemble_L2 | extreme | 0.1268 | **0.0491** |
| 4 | 11 | TabM_r184_BAG_L1 | extreme | 0.1266 | **0.0781** |
| 5 | 9 | XGBoost_BAG_L1 | best | 0.1259 | **0.0795** |
| 6 | 5 | NeuralNetFastAI_BAG_L1 | best | 0.1131 | **0.0890** |
| 7 | 4 | NeuralNetFastAI_r102_BAG_L1 | best | 0.1098 | **0.0944** |
| 8 | 14 | NeuralNetTorch_r22_BAG_L1 | best | 0.1344 | **0.0956** |
| 9 | 2 | NeuralNetFastAI_r191_BAG_L1 | best | 0.0985 | **0.0987** |
| 10 | 20 | TabM_r191_BAG_L1 | extreme | 0.1504 | **0.1000** |
| OPT1 + OPT2 → REF | | | | | |
| 1 | 10 | TabM_r52_BAG_L1 | extreme | 0.5992 | **0.0587** |
| 2 | 17 | WeightedEnsemble_L2 | extreme | 0.6290 | **0.0801** |
| 3 | 47 | TabM_r184_BAG_L1 | extreme | 0.6548 | **0.1050** |
| 4 | 5 | NeuralNetTorch_r79_BAG_L1 | best | 0.5585 | **0.1304** |
| 5 | 2 | NeuralNetFastAI_BAG_L1 | best | 0.4332 | **0.1365** |
| 6 | 1 | NeuralNetFastAI_r191_BAG_L1 | best | 0.4228 | **0.1443** |
| 7 | 3 | NeuralNetFastAI_r102_BAG_L1 | best | 0.4903 | **0.1502** |
| 8 | 15 | TabM_r191_BAG_L1 | extreme | 0.6155 | **0.1564** |
| 9 | 4 | NeuralNetTorch_BAG_L1 | best | 0.5384 | **0.1705** |
| 10 | 16 | TabM_BAG_L1 | extreme | 0.6210 | **0.1732** |

## M CAN THE LABELS BE REPRODUCED FROM THE RAW SIGNALS?

The full $\approx 190$ GB OpenFAST time-series record is not part of the first release (Section 6), so the label pipeline (bending moment $\rightarrow$ stress $\rightarrow$ rainflow $\rightarrow$ S–N $\rightarrow$ Miner $\rightarrow$ DEL) cannot be audited end to end from the dataset alone. We therefore release a raw time-series audit subset (anonymized link in Appendix B) with a runner that regenerates the released labels from the signals.

**Composition and contents.** The subset is `sim_id` 1 to 49 on each tower: the complete $7 \times 7$ wave grid at the cut-in wind level (itself an extrapolation level), turbulence seed S1, and the same operating conditions on all three towers. It includes near-zero damage labels, where rainflow counting is most sensitive to numerical tolerance. The 147 outputs take $\approx 1.4$ GB (9.3 MB each) and are released under CC-BY-4.0. The package holds the raw OpenFAST outputs, their labels, the tower geometries, the post-processing code (stress, rainflow, DNV-RP-C203 curve E with thickness correction, Miner, DEL), the reference Python environment, and the audit runner.

**Result.** The acceptance criterion is a per-label relative deviation $|\hat{D} - D| / \max(|D|, 10^{-12}) \leq 10^{-6}$, which leaves headroom for platform-dependent floating-point differences. On all 147 simulations the pipeline reproduces the 4,410 released labels with a maximum relative deviation of $4.6 \times 10^{-16}$, i.e. machine precision.

**Chain of custody.** The OpenFAST version (v3.5.2, double precision) is recorded in every output file, and the tower input decks are part of the released geometry artifacts. The subset audits the chain time series $\rightarrow$ labels end to end; re-running a simulation from scratch also needs its turbulence and wave seeds, which the subset does not include. Lifetime (weighted) damage needs no signals: it follows from the released labels and `damage_weight` (Appendix E).

**Summary.** Yes, on the released audit subset. On all 147 simulations the pipeline reproduces the 4,410 released labels to machine precision (maximum relative deviation $4.6 \times 10^{-16}$). Re-running a simulation from scratch also needs its turbulence and wave seeds, which the subset does not include.

# N DATASHEET FOR DATASETS

This appendix follows the Gebru et al. (2021) template. Headings preserved verbatim where possible; full Q&A reproduced below.

## 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.** FLOATBench was created to serve as a public, benchmark-quality reference for tabular machine-learning surrogates of per-section fatigue damage on floating offshore wind turbine (FOWT) towers, with explicit boundary-reliability evaluation under a regime-aware partition. The specific task in mind is regression of per-section fatigue damage as a function of the joint wind/wave operating envelope. The specific gap is that prior work in this domain reports per-paper simulations, splits, and metrics (Table 1), so the field cannot adjudicate between competing surrogates on common ground. The release supports both engineering re-use (training surrogates inside FOWT design pipelines) and ML re-use (anchoring tabular surrogate comparisons on a physics-grounded regression target).

**Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?** The dataset was created by the authors of this paper; their identities and affiliations are withheld for anonymous review.

**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). No grant was earmarked specifically for FLOATBench.

**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 released row represents the (fatigue-relevant) outcome of one ten-minute aero-hydro-servo-elastic OpenFAST simulation projected onto one of 30 cross-sections of one of three FOWT tower geometries. Features are environmental (`wind_speed`, `mean_wind_speed`, `std_wind_speed`, `wave_hs`, `wave_tp`, `wind_seed_id`) and structural (`section_height_m`, `section_radius_m`, `section_thickness_m`). The released target is `damage` (Miner-summed fatigue damage at the section, per the schema in Table 6); the DEL-format target used in the benchmarks is the cube-root damage transform $D^{1/3}$, computed from `damage` on demand. Lifetime aggregation uses `damage_weight` (expected number of 600 s windows over the 25-year service life; Appendix E) so that the lifetime damage at a section is recovered as $\sum_i \texttt{damage}_i \cdot \texttt{damage_weight}_i$ over all conditions.

**How many instances are there in total (of each type, if appropriate)?** $582{,}120$ rows in aggregate ($3$ towers $\times$ $6{,}468$ simulations $\times$ $30$ sections; the released grid contains $22 \times 7 \times 7 = 1{,}078$ operating points per tower, with 6 seeds per point and 30 sections per tower).

**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)?** The released grid is a deliberate sample of a larger continuous operating envelope: $22 \times 7 \times 7 = 1{,}078$ cartesian (wind, $H_s$, $T_p$) operating points per tower, with 6 turbulence seeds per point. The grid spans the joint wind/wave envelope encountered by an aligned-wind–wave FOWT under DLC 1.2 (normal power production), from cut-in to cut-out wind speed under IEC class A turbulence and a JONSWAP $\gamma = 3.3$ wave spectrum (JONSWAP is the standard irregular-wave spectrum recommended by IEC 61400-3-2 for offshore design; $\gamma$ is its peak-enhancement factor). The dataset is closed under DLC 1.2 and aligned-only wind–wave assumptions; non-aligned conditions, parked or idling DLCs, and fault DLCs are out of scope.

**What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.** Features. Each row is the Miner-summed fatigue damage at one tower section for one simulation. It is extracted by a custom rainflow counting implementation from a 600-second post-burn-in window of the section bending-moment series. The DEL-format target $D^{1/3}$ is derived on demand and not stored. Raw OpenFAST time series are not redistributed, except the 147-simulation audit subset (Appendix M).

**Is there a label or target associated with each instance? If so, please provide a description.** Yes. The released target column is `damage` (Miner-summed fatigue damage at the section); the DEL-format target used in the benchmarks is the cube-root damage transform $D^{1/3}$, computed from `damage` on demand. See schema in Table 6.

**Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing.** No.

**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. `sim_id` ties together the 30 cross-sections of the same simulation, and `wind_seed_id` ties together the 6 random-seed realizations of the same operating point.

**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: the regime-aware alpha-shape partition is the recommended split (Section 4.1), exposed by the released splitter. The rationale is that a random split collapses the test set onto the in-train interior of the operating envelope and thereby hides the boundary-extrapolation behavior that drives certification-grade use; the regime-aware partition exposes nine wind×wave regime cells, including the deepest wind-and-wave extrapolation cell.

**Are there any errors, sources of noise, or redundancies in the dataset? If so, please provide a description.** Aleatoric noise from turbulence and wave-elevation realization is the dominant source. Its spread can be computed from the six seeds per condition. Numerical noise from OpenFAST integration is sub-dominant. The simulation setup is in Section 3 and the label pipeline audit in Appendix M.

**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 as released. The upstream OpenFAST inputs (turbine, controller, platform) are linked back to the IEA Wind Task 55 release (Zahle et al., 2024), which is hosted on a public GitHub repository under Apache-2.0 with version-tagged releases; the IEA release is not under our control, but its versioned public hosting provides a stable archival anchor for re-running the upstream simulations if needed.

**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 physics simulations, 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.** Indirectly derived from physics simulation: each instance is the output of an aero-hydro-servo-elastic OpenFAST run under DLC 1.2 conditions (NTM, NWP, IEC class A, JONSWAP $\gamma = 3.3$), post-processed via rainflow counting and Miner summation against the DNV-RP-C203 curve E S–N curve (bilinear, $m = 3/5$, with thickness correction). Validation: OpenFAST is a peer-reviewed NREL solver with extensive code-to-code and code-to-experiment validation in the open literature; our tower-loads post-processing pipeline is released, and its audit runner reproduces the released labels from the raw OpenFAST outputs to machine precision (Appendix M). See Section 3 for the full setup.

**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?** OpenFAST (Jonkman, 2013) (the NREL open-source aero-hydro-servo-elastic simulator) executed in batch on a commercial cloud HPC platform; rainflow counting via a custom in-house implementation; Miner summation against a DNV-RP-C203 (DNV, 2024) S–N curve through a custom Python pipeline released alongside the dataset. The OpenFAST setup follows the IEA Wind Task 55 (Zahle et al., 2024) reference release, and the rainflow + S–N + Miner pipeline is released with an audit runner that reproduces the labels from the raw outputs (Appendix M).

**If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?** Deterministic cartesian grid: 22 wind speeds (cut-in to cut-out, 1 m/s spacing) $\times$ 7 significant wave heights $H_s$ $\times$ 7 peak periods $T_p$, giving 1,078 operating points per tower, and 6 i.i.d. turbulence seeds per point.

**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; no crowdworkers or paid annotators. Compensated through their academic positions.

**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.** Three OpenFAST simulation campaigns, one per tower geometry, followed by post-processing, labeling, and benchmark preparation, all completed before submission. 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 PII.

**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 human-subjects data.

**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: 400-second start-up burn-in discarded; rainflow counting per section (custom in-house implementation) followed by Miner summation against the DNV-RP-C203 curve E S–N curve yields the released `damage` target $D$. The DEL-format target $D^{1/3}$ used in the benchmarks is computed from `damage` on demand (not stored as a separate column).

**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.** Raw OpenFAST output is not re-released for storage reasons; the deterministic preprocessing pipeline is released as code (MIT-licensed). A raw time-series audit subset of 147 simulations, with a runner that regenerates the released labels from the signals, is released (Appendix M).

**Is the software that was used to preprocess/clean/label the data available? If so, please provide a link or other access point.** Yes, released alongside the dataset; the code repository is referenced in the paper's URL appendix (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 82 to 96 AutoGluon models per tower in E1/E2 and 58 to 63 per fold in E3 (735 in total) plus seven standalone baselines under the regime-aware split protocol; see Section 5.

**What (other) tasks could the dataset be used for?** (i) tabular regression surrogate benchmarking; (ii) tabular surrogate boundary-reliability benchmarking; (iii) physics-grounded transfer-learning across geometries; (iv) multi-task expansion (e.g. auxiliary prediction of per-section peak stress, top deflection, buckling utilisation ratios) once those targets are released in a follow-up.

**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.** Will be maintained on the dataset repository page (anonymized link 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 individuals 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. The dataset is restricted to (i) DLC 1.2 (normal power production), (ii) aligned wind–wave only, (iii) IEC class A turbulence, (iv) JONSWAP $\gamma = 3.3$ wave spectrum, (v) 600 s analysis window per condition after burn-in, and (vi) the DNV-RP-C203 curve E S–N curve (bilinear, $m = 3/5$). Surrogates trained on FLOATBench should not be expected to generalize without retraining or transfer-learning to misaligned wind–wave conditions, parked or idling DLCs, fault-induced loads, or alternative S–N classes. Dataset consumers planning to use FLOATBench for certification-adjacent decisions must couple it with the full DLC matrix specified in IEC 61400-3-2 (IEC, 2019).

**Are there tasks for which the dataset should not be used? If so, please provide a description.** Yes: the dataset is *not certification-grade* and should not substitute for full DLC analysis in regulatory contexts.

**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 via 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)?** Three CSVs per tower (training set, test set, and raw rows), hosted at an anonymized link (Appendix B) during review; a Croissant 1.1 metadata JSON and a DOI will be released with the final version.

**When will the dataset be distributed?** At the time of submission and with the final version (final DOI).

**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/`); MIT for the accompanying code. 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.** The IEA-22-280-RWT reference turbine and floating support structure (Zahle et al., 2024) on which the simulation campaign is built is released under Apache-2.0; FLOATBench is a downstream derivative and inherits no usage restriction beyond Apache-2.0 attribution. No other third party imposes restrictions on the released CSVs.

**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.** None.

**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 dataset repository page.

**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 through the dataset repository version history and a corresponding `ERRATA.md` file in 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. Updates will be issued by the authors as needed (no fixed cadence) and will be communicated through the dataset repository page (which includes a public commit history) and a release-notes entry per version. The authors commit to a v1.x release accompanying the paper and to logging changes in the repository. As additional turbine families are simulated, cross-design transfer splits will be published as new versioned releases; the regime partition depends only on the wind/wave grid and is reused unchanged.

**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. All released versions remain hosted on the dataset repository indefinitely with version-pinned access (pinned revisions), so downstream papers that pin to a specific release (e.g. v1.0) remain reproducible after later releases.

**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?** The dataset repository will host community CSV contributions (additional towers, additional DLCs) under the same CC-BY-4.0 license with attribution. Contributions are accepted as additional versioned releases, not as overrides of v1.x; the authors will spot-check contributed CSVs against the released schema and will flag (rather than reject) contributions whose simulation provenance is not reproducible from the linked code.

**Any other comments?** None.

## O AI USE 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 partition-sensitivity study, the alternative training grids and the standalone baselines, to support data analysis and the interpretation of results, to refine the framing of the benchmark, and to clean and reformat the dataset for release. We have not used generative AI tools to generate data. Every label comes from an OpenFAST simulation run by the authors, and the tools only helped write the post-processing scripts. 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, summarize and check related work and format references. We have reviewed all AI-assisted work. Every number in the paper was re-derived from the released dataset and trained surrogates by the released evaluation and bootstrap 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.