A Strut and Tie Neural Network Surrogate for Failure-Load Prediction of Concrete D-Regions Designed by the Compatible Stress Field Method

Sandesh Lamsal* Rubi Bhandari

AtkinsRéalis USA Inc., 800 Waterford Way, Miami, FL 33126, USA
Department of Civil and Architectural Engineering, University of Miami, Coral Gables, FL 33146, USA
Department of Civil and Environmental Engineering, Florida International University, Miami, FL 33174, USA

Abstract

The Compatible Stress Field Method (CSFM) designs the discontinuity regions (D-regions) of structural concrete by combining a lower-bound stress field with kinematic compatibility and realistic constitutive behaviour. Its nonlinear solution procedure is, however, too slow for the repeated evaluations demanded by design-space exploration, reinforcement optimisation and reliability analysis. This study presents a neural-network surrogate that predicts the failure load factor of a D-region directly from its design parameters. A multilayer perceptron maps a normalised description of a design directly to the failure load factor and the strut and tie (STM) member forces at the failure state, so the surrogate returns an interpretable force state, not the strength alone. Training data come from sweeping an existing CSFM solver over a Latin Hypercube design of experiments across four D-region archetypes: deep beam, hammerhead pier, multi-column bent and pile cap. Trained per archetype, the surrogate predicts the failure load factor with a coefficient of determination of 0.95–0.99 and a mean absolute percentage error of 3–5 %, far more accurately than a gradient-boosted-tree baseline and at roughly three orders of magnitude lower cost than the reference solver. The reference solver is itself checked against an experimental pier-cap benchmark, completing a two-tier validation, though that tier rests on a single five-specimen series. A bagged deep ensemble with split-conformal calibration equips each prediction with an interval that has a finite-sample coverage guarantee. The surrogate makes the design-space exploration, reinforcement optimisation and reliability analysis of D-regions tractable.

Keywords: Compatible Stress Field Method, strut and tie model, discontinuity regions, neural-network surrogate, machine learning, surrogate model, structural concrete, failure load.

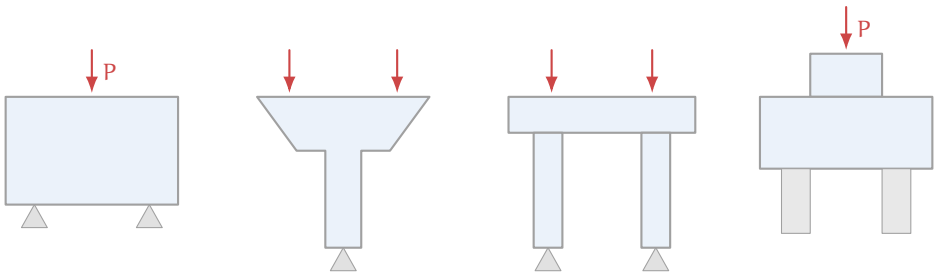
1 Introduction

Reinforced-concrete structures contain regions in which the Bernoulli hypothesis of plane sections does not hold: the vicinity of concentrated loads and supports, frame corners, corbels, deep beams, openings, pile caps and hammerhead piers. In these *discontinuity regions* (D-regions) the strain distribution is markedly nonlinear, and design must follow the actual flow of forces rather than sectional analysis. Strut and tie modelling (STM), placed on a consistent footing for design practice by Schlaich et al. [15], idealises this flow as a pin-jointed truss of concrete struts and reinforcement ties and applies the lower-bound theorem of plasticity to deliver a safe design.

Classical STM is powerful but has well-known limitations: the truss topology is not unique and relies on engineering judgement, serviceability (deformations and crack widths) is not addressed, and the verification of nodal zones is often conservative and laborious. The Compatible

*Corresponding author: sandeshlamsal@miami.edu

Stress Field Method (CSFM), developed by Kaufmann et al. [6] and now embedded in commercial design software, removes these limitations by combining a continuous (or discretised) stress field with kinematic compatibility and finite, realistic material laws: a parabola-rectangle law with compression softening for concrete [2] and a bilinear law with tension stiffening for reinforcement. The CSFM thereby yields not only a safe strength verification but also the load–deformation response, the failure load and the governing failure mode. Figure 1 illustrates the four D-region archetypes considered in this study.



(a) Deep beam (b) Hammerhead pier (c) Multi-column bent (d) Pile cap

Figure 1: The four discontinuity-region (D-region) archetypes covered by the dataset. Each is idealised as a strut and tie model and analysed by the Compatible Stress Field Method to generate ground-truth failure load factors.

The price of this richness is computational cost. A CSFM analysis is an iterative nonlinear procedure: the reference load set is scaled by an increasing load factor, and at each increment the constitutive nonlinearity, in particular the strain-dependent softening of cracked concrete, is resolved by inner iterations. A single design evaluation therefore involves many tens of linear solves. This is acceptable for the verification of one design, but it becomes the bottleneck whenever *many* designs must be evaluated: in parametric studies, reinforcement-layout optimisation, or probabilistic reliability assessment, where the same nonlinear analysis is repeated thousands of times.

Machine-learning surrogates are an obvious remedy, and data-driven models of structural response have proliferated. A surrogate that, once trained, returns the failure load of a D-region in a single forward pass would make these many-evaluation workflows tractable, provided it is accurate enough to be trusted and is validated against the solver it replaces.

This study presents a neural-network surrogate of the CSFM. The surrogate maps a D-region's design parameters directly to its failure load factor and to the strut and tie member forces at the failure state, so that it returns a physically interpretable force state alongside the strength rather than the strength alone. The contributions of this study are:

1. *A neural-network surrogate* that predicts the failure load factor and the failure-state strut and tie forces of CSFM D-regions directly from their design parameters.

2. *A demonstration across four D-region archetypes* (deep beam, hammerhead pier, multi-column bent and pile cap), with a coefficient of determination of 0.95–0.99, far above a gradient-boosted-tree baseline, and roughly three orders of magnitude speed-up over the reference solver.

3. *A two-tier validation*: the surrogate is verified against the CSFM solver, and the solver is in turn checked against an experimental pier-cap benchmark from the literature, a single but well-documented five-specimen series.

4. *Calibrated prediction intervals* from a bagged deep ensemble with split-conformal calibration, attaching to each prediction an interval with a finite-sample coverage guarantee.

5. *An automated data-generation pipeline* that sweeps an existing CSFM solver over a Latin Hypercube design of experiments, released together with the trained models and the evaluation code (see the Data and code availability statement).

The surrogate replaces the reference solver only where many evaluations are needed; its accuracy ceiling is that of the solver, which is checked against experiments in Section 4. This study is the first in a planned series on machine-learning methods for the CSFM. The remainder of this study is organised as follows. Section 2 reviews STM, the CSFM and machine-learning surrogates. Section 3 develops the surrogate. Section 4 validates the reference solver against physical experiments. Sections 5 and 6 report and discuss the results. Section 7 states the limitations and Section 8 concludes.

2 Related work

Strut and tie modelling and stress fields. The strut and tie method for the design of D-regions was placed on a consistent footing by Schlaich et al. [15], who unified B-region and D-region design under the lower-bound theorem of plasticity. The plasticity basis of truss and stress-field design was developed by Marti [10]. Stress-field methods generalise the discrete truss to continuous fields of compression and tension [12]. The CSFM, as formalised by Kaufmann et al. [6], augments the stress field with kinematic compatibility and finite, realistic constitutive laws, including the compression softening of cracked concrete; it delivers load–deformation behaviour, failure loads and crack widths within a single, code-oriented framework, and it underpins the software implementation used here as the data reference CSFM solver. The compression softening on which the method depends traces back to the modified compression-field theory of Vecchio and Collins [16], which established that the compressive strength of concrete falls as transverse cracking increases; the CSFM carries that effect into a stress-field setting. As a method it occupies a middle ground between the discrete strut and tie model, which a designer lays out by hand, and a full nonlinear finite-element analysis: it keeps the transparency of the former while supplying the deformation capacity and failure load of the latter. That intermediate character is what makes it a suitable reference CSFM solver for a surrogate, since each analysis is informative yet still individually inexpensive.

Physics-informed and physics-constrained neural networks. Physics-informed neural networks (PINNs), popularised by Raissi et al. [13], embed the residual of a governing differential equation into the loss of a neural network, so that the network is trained to satisfy the physics with little or no labelled data; the broader programme is surveyed by Karniadakis et al. [5]. A complementary strand, the deep energy method of Samaniego et al. [14], replaces the strong-form residual by the system's potential energy. Training such networks is not always straightforward: the residual loss can be stiff and ill-conditioned, and Wang et al. [17] document gradient-flow pathologies that stall convergence when a physics term competes with data terms. These difficulties are most pronounced where the governing relations are non-smooth, as the piecewise constitutive laws of the CSFM are. Most of this literature also targets continuous domains and uses automatic differentiation to evaluate PDE residuals, whereas a D-region analysed by the CSFM is naturally posed on a discrete strut and tie graph. The present study therefore does not use a PINN; it trains a purely supervised surrogate of the CSFM solver, and treats the solver itself as the carrier of the physics.

Machine-learning surrogates in structural engineering. Data-driven surrogates have been applied widely to structural-response prediction, capacity estimation and design optimisation, built variously on response surfaces, Gaussian-process regression, tree ensembles and neural networks. In the specific context of D-regions, machine learning has mostly been used either to predict a single scalar capacity, such as the shear strength of a deep beam, or to assist the selection of a strut and tie topology. Two features distinguish the surrogate proposed here. It predicts not only the failure load factor but also the failure-state strut and tie member forces,

so its output is a complete and physically interpretable design state rather than an opaque capacity number. And it targets the CSFM, a compatibility-based method, so the labels it learns from already embed deformation capacity and compression softening rather than an empirical strength formula. The surrogate is validated in two tiers: against the solver it emulates and, through that solver, against physical experiments.

Uncertainty quantification for surrogates. A surrogate intended for reliability analysis must report not only a prediction but a measure of how far that prediction can be trusted. Deep ensembles, introduced by Lakshminarayanan et al. [8], estimate this by training several networks and reading the spread of their predictions as an epistemic-uncertainty signal; they need no change to the network and are a standard baseline for this purpose. Their raw spread is not guaranteed to be calibrated, however. Conformal prediction [1, 9] supplies the missing guarantee: from a held-out calibration sample it produces intervals whose coverage holds in finite samples, with no distributional assumption. The present study combines the two, using the ensemble spread to shape the interval and conformal calibration to certify it, in Section 5.4.

3 Methodology

3.1 The Compatible Stress Field Method on the strut and tie graph

A D-region is represented by a space truss. Let $\mathcal{N}$ be the set of nodes, each with position $\mathbf{x}_i \in \mathbb{R}^3$ and prescribed support conditions, and $\mathcal{M}$ the set of members, each connecting two nodes and carrying an axial force only (tension positive). A reference load set applies nodal forces $\mathbf{P}_i$, scaled by a scalar load factor λ. Each member is assigned a concrete cross-sectional area $A_{c,m}$ and a reinforcement area $A_{s,m}$, so that it may act as a strut or as a tie depending on the sign of its strain. Figure 2 shows the strut and tie graph of a deep-beam D-region.

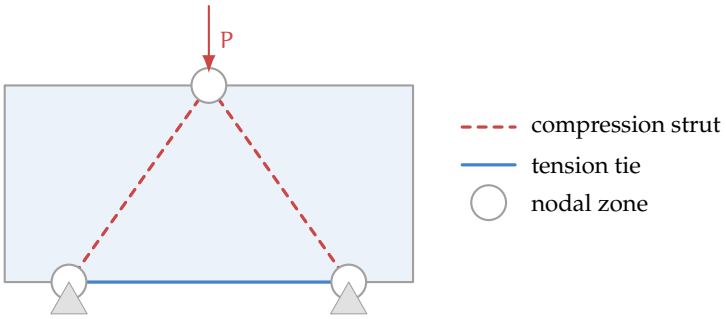


Figure 2: Strut and tie graph of a deep-beam D-region under a central load P. The load spreads through inclined concrete struts (compression) to the supports, and a bottom reinforcement tie (tension) carries the horizontal thrust. The surrogate predicts the failure load factor and the member forces of this graph.

Kinematics. Let $\mathbf{u}_i \in \mathbb{R}^3$ be the displacement of node i. For a member m connecting nodes a and b, with unit axial vector $\mathbf{c}_m$ and length L_m, the axial elongation and average strain are

$$e_m = (\mathbf{u}_b - \mathbf{u}_a) \cdot \mathbf{c}_m, \qquad \varepsilon_m = e_m / L_m. \tag{1}$$

Constitutive laws. Concrete in compression follows a parabola-rectangle law [2], with the peak strain ε_{c2}, ultimate strain ε_{cu2} and exponent n taken as functions of the characteristic strength f_{ck}. The effective compressive strength of *cracked* concrete is reduced by a compression-softening factor k_{c2} that decreases with the principal (transverse) tensile strain ε_1, and by a brittleness factor η_{fc}. In the discrete truss ε_1 is not available pointwise; it is estimated from the strains of the reinforcement crossing the strut and the effective reinforcement ratio, consistent with the host implementation. Reinforcing steel follows an idealised bilinear law with a yield plateau at f_y and a hardening branch to (ε_u, f_t). The two laws are plotted in Figure 3.

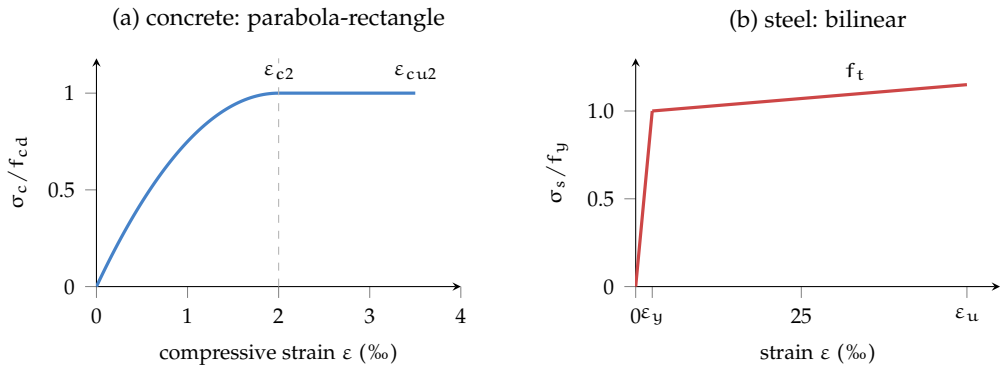


Figure 3: Constitutive laws of the Compatible Stress Field Method, as evaluated by the reference solver: (a) the parabola-rectangle law for concrete in compression, with peak and ultimate strains and the parabola exponent taken as functions of f_{ck}, the effective strength further reduced by the compression-softening factor k_{c2}; (b) the idealised bilinear law for reinforcing steel. Axes are normalised; strain on a different scale in each panel.

Member forces and the failure criterion. Given the strain ε_m, the member force is

$$F_m = \begin{cases} -A_{c,m}\,\sigma_c(|\varepsilon_m|, \varepsilon_1)\,, & \varepsilon_m < 0 \quad \text{(strut action)}, \\ +A_{s,m}\,\sigma_s(\varepsilon_m)\,, & \varepsilon_m \geqslant 0 \quad \text{(tie action)}, \end{cases} \tag{2}$$

with σ_c and σ_s the concrete and steel laws above. A member utilisation U_m is the ratio of its force to its capacity, taken as the effective concrete compressive resistance for struts and the yield force for ties. The failure load factor λ_f is the smallest load factor at which the maximum member utilisation reaches unity.

3.2 Dataset generation

The dataset is produced by an automated sweep of the reference CSFM solver. Four D-region archetypes are considered: a deep beam, a hammerhead pier, a multi-column bent and a pile cap. For each archetype the free parameters are the principal geometric dimensions, the concrete grade f_{ck}, the steel grade f_y, the provided reinforcement (area and bar diameter) and the reference load. Designs are drawn by Latin Hypercube Sampling [11] over the joint design space, which gives even space-filling coverage with far fewer samples than a full grid. Samples that the solver reports as geometrically unstable or otherwise invalid are discarded and resampled. Each accepted design is analysed by the reference solver; the recorded quantities are the strut and tie geometry and connectivity, the member forces at the static solution, the failure load factor, the failure mode and the load–displacement curve. The failure load factor is the principal label; the member forces support the optional supervised anchor term. Designs are split into training, validation and test sets *by design*, not by load factor, so that no test design is seen during training; a separate extrapolation set is generated outside the training ranges to probe generalisation. The pipeline exports the dataset as JSON consumed by the training code, decoupling data generation from learning. Figure 4 summarises the workflow.

For each archetype, 750 valid designs are generated. The fixed-size network requires a single strut and tie topology per archetype; the minority of sampled geometries that flip a truss diagonal form a secondary topology and are set aside, which in practice affects only the hammerhead. The retained designs are split 70/15/15 by design. Table 1 reports the resulting counts: each network is trained on 378 to 525 designs and tested on 81 to 113. The per-archetype accuracy of Section 5 is reported on the subset of test designs that fail within the analysed load range, which is smaller again; the full test split, not that subset, is the basis for generalisation.

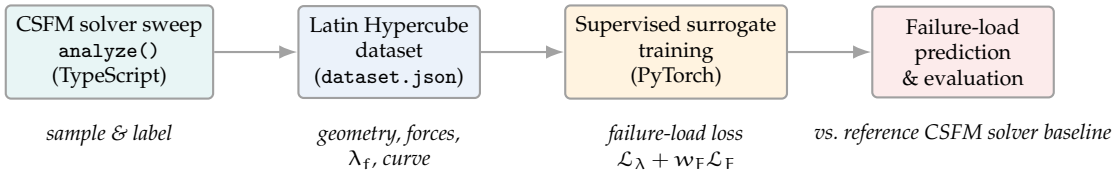


Figure 4: Data-generation and training pipeline. An existing Compatible Stress Field Method solver is swept over a Latin Hypercube design of experiments to produce a labelled dataset; the surrogate network is trained in PyTorch by supervision on the failure load factor and the failure-state member forces; the trained surrogate is evaluated against the reference solver as the ground truth.

Table 1: Per-archetype dataset sizes. *Generated* is the number of valid Latin-hypercube designs analysed by the reference solver; *retained* is the subset sharing the dominant strut and tie topology, split by design into training, validation and test sets.

Archetype	Generated	Retained	Train	Val.	Test
Deep beam	750	750	525	112	113
Hammerhead	750	540	378	81	81
Multi-column bent	750	750	525	112	113
Pile cap	750	750	525	112	113

3.3 Network and direct failure-state prediction

Let a D-region design be described by a normalised parameter vector θ collecting geometry, material properties ($f_{ck}, f_y, E_s, \ldots$), the provided reinforcement and the load pattern. The surrogate is a multilayer perceptron that maps θ *directly* to the two quantities of interest:

$$\mathcal{S} : \theta \longmapsto \left(\lambda_f,\ \mathbf{F}^f \in \mathbb{R}^{|\mathcal{M}|}\right), \tag{3}$$

the failure load factor λ_f and the member forces $\mathbf{F}^f$ at the failure state. An earlier formulation that predicted the full nodal displacement field at every load level and recovered λ_f by a search over the load factor proved fragile: the failure load is sensitive to small errors in the predicted field, and the search amplifies them. Predicting the failure state directly removes that indirection. The failure load factor is passed through a softplus so that it is positive; the member forces are unconstrained (tension positive). Inputs are standardised to zero mean and unit variance over the training split. The strut and tie topology is fixed within an archetype, so one network is trained per archetype. The architecture is a multilayer perceptron with six hidden layers of width 128 and tanh activations; with an input dimension of 9–10 design parameters and an output dimension of $1 + |\mathcal{M}|$, this amounts to roughly 85,000–88,000 trainable parameters depending on the archetype. The architecture and training settings are collected in Table 2, and Figure 5 shows the network.

3.4 Loss

The network is trained by supervision against the reference CSFM solver. The primary term is the squared error on the failure load factor,

$$\mathcal{L}_\lambda = \left(\lambda_f - \lambda_f^{\text{ref}}\right)^2, \tag{4}$$

averaged over the design batch. A second term anchors the predicted failure-state member forces to those computed by the reference CSFM solver,

$$\mathcal{L}_F = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \left(\frac{F_m^f - F_m^{f,\text{ref}}}{F_0} \right)^2, \tag{5}$$

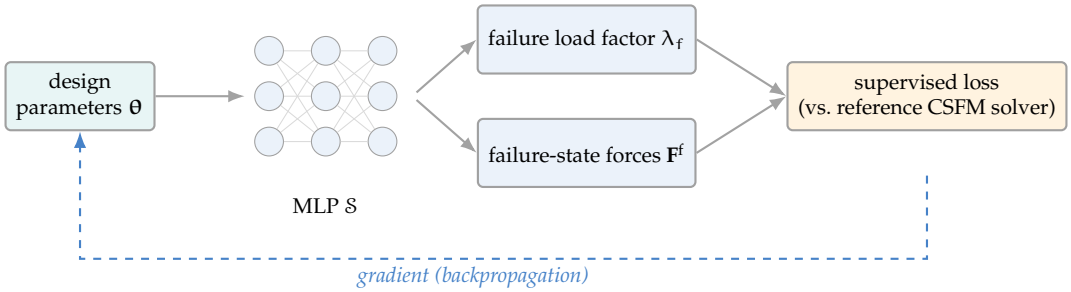


Figure 5: The neural-network surrogate. A multilayer perceptron maps the normalised design parameters θ directly to the failure load factor λ_f and the strut and tie member forces $\mathbf{F}^f$ at the failure state. Both outputs are trained by supervision against the reference Compatible Stress Field Method solver.

with F_0 a per-design force scale, so that the surrogate returns a physically interpretable strut and tie force state alongside the failure load factor. The total loss is

$$\mathcal{L} = \mathcal{L}_\lambda + w_F \mathcal{L}_F, \tag{6}$$

with w_F a fixed weight.

3.5 Training

One network is trained per archetype. The designs are split into training, validation and test sets *by design* (Section 3.2); training minimises Eq. (6) over mini-batches with AdamW [7] under a cosine-annealed learning rate, with gradient-norm clipping. The checkpoint with the lowest failure-load validation loss is retained. All archetypes share the architecture and training settings of Table 2; these were fixed early and not tuned per archetype, so the per-archetype accuracy differences of Section 5 reflect the data rather than hyperparameter search. Because the network outputs λ_f directly, evaluation is a single forward pass that requires no load-factor sweep and no constitutive evaluation; this is the source of the speed-up over the reference CSFM solver reported in Section 5.

Table 2: Network architecture and training hyperparameters, shared by all four archetypes.

Setting	Value
Hidden layers	6
Hidden width	128
Activation	tanh
Input dimension	9–10 design parameters
Output dimension	$1 + \lvert\mathcal{M}\rvert$
Trainable parameters	85,000–88,000
Optimiser	AdamW
Learning rate	10^{-3}, cosine-annealed
Weight decay	10^{-5}
Gradient-norm clip	5.0
Batch size	64 designs
Epochs	400
Force-loss weight w_F	1.0
Train / validation / test	70 / 15 / 15 % by design

4 Validation of the reference solver

The surrogate of Section 3 is trained to reproduce the reference CSFM solver; its predictions are therefore only as physically meaningful as that solver. This section examines the solver, hereafter the *reference solver*, against physical experiments. The validation has two tiers: the surrogate is verified against the solver (Section 5), and the solver is checked here against measured failure loads. The second tier rests on a single, well-documented experimental series and probes the reference solver for one representative D-region type rather than exhaustively across the design space.

4.1 Experimental benchmark

The benchmark is the series of reinforced-concrete pier-cap tests of Geevar and Menon [4], analysed with the CSFM by Kaufmann et al. [6]. Five specimens (S1–S5) with constant geometry and concentric loading are considered. Each is a stepped D-region comprising a wide cap band, a tapered transition and a narrow stem, loaded through a bearing plate on the stem and reacted by four supports under the cap; all five failed in the tests by concrete crushing of the diagonal strut. Following the validation convention of [6], the mean measured material properties are used with no partial safety factors ($\gamma = 1$). The reference solver already evaluates the failure load at nominal material strength, so this requires only that mean properties be supplied as input.

4.2 Modelling and results

A geometrically faithful model is essential. When the stepped specimen is idealised as a prismatic block, the diagonal-strut concrete area is over-estimated, the strut never reaches its crushing strength, and the solver incorrectly predicts a reinforcement-rupture failure for every specimen. Modelling the true stepped outline, with the strut concrete area taken from the real confined section (the out-of-plane thickness times the loading-plate width), restores the correct mechanism: the solver predicts concrete crushing for four of the five specimens, consistent with the tests.

Table 3 compares the measured ultimate load P_{exp} with the solver prediction P_{calc} (P is the total applied load), and Figure 6 shows the comparison per specimen against the measured loads and the continuum CSFM analysis of the book.

Table 3: Reference-solver validation against the pier-cap tests of Geevar and Menon, with mean material properties and $\gamma = 1$.

Specimen	f_c (MPa)	P_{exp} (kN)	P_{calc} (kN)	P_{exp}/P_{calc}	failure mode
S1	31.2	2224	1890	1.18	crushing
S2	35.7	3068	1994	1.54	rupture
S3	30.1	3436	2233	1.54	crushing
S4	34.9	3608	2526	1.43	crushing
S5	34.1	3464	2771	1.25	crushing
mean / CoV				1.39 / 0.11	

The solver reproduces the experimental failure loads with low scatter, a coefficient of variation of 0.11 against the 0.13 of the continuum CSFM analysis reported in [6], but with a systematic conservative bias: the mean ratio $P_{exp}/P_{calc} = 1.39$, so the predicted capacity is about 72 % of the measured value.

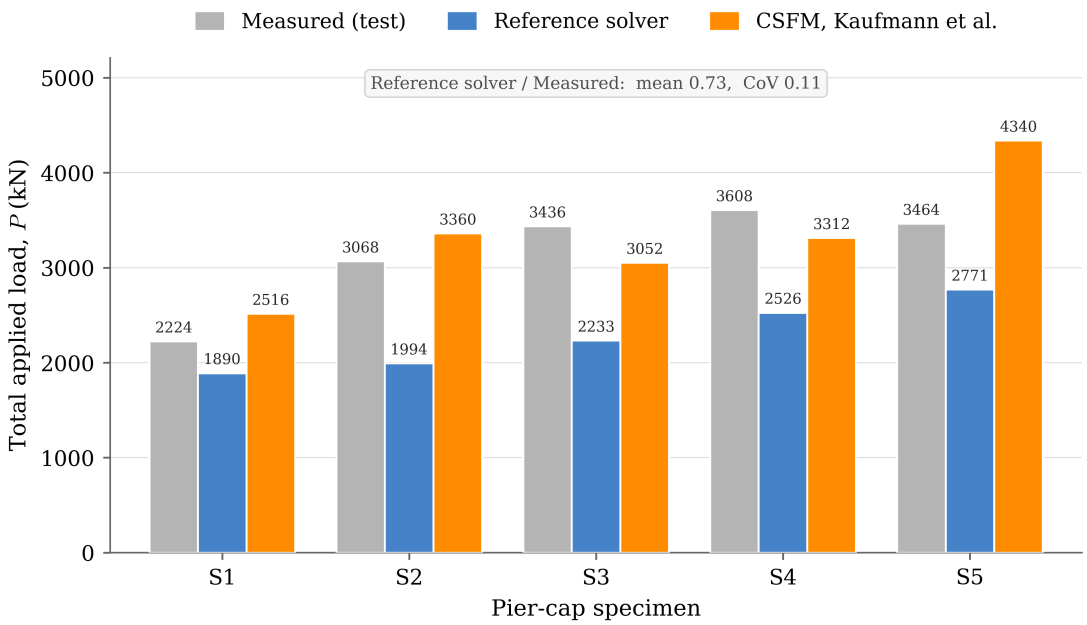


Figure 6: Reference-solver validation against the five Geevar and Menon [4] pier-cap specimens: the measured ultimate load, the reference solver's prediction, and the continuum CSFM analysis of Kaufmann et al. The solver tracks the experiments consistently (coefficient of variation 0.11) with a systematic conservative bias.

4.3 Discussion

The bias is a known consequence of representing a D-region by a truss with *fixed* member cross-sections. A real strut spreads transversely, forming a so-called "bottle" strut, and a continuum stress-field analysis, such as the reference analysis of [6], resolves this spreading. A truss strut of fixed area, sized at the loaded nodal zone, instead under-estimates the effective strut and hence the crushing load. The low scatter (CoV 0.11) confirms that the model is mechanically consistent up to this scale factor; a continuum stress-field solver was also evaluated but produced spurious early failures at the re-entrant corners of the stepped outline and is not reported.

Two implications follow. First, the reference solver is a sound basis for the surrogate: it predicts the correct failure mode and ranks the specimens with low scatter, so a surrogate trained on it learns physically consistent behaviour. Second, the conservative bias for strut-crushing-governed D-regions is a genuine limitation of the discrete strut and tie formulation that is addressed in subsequent work.

4.4 Direct validation of the surrogate

The two-tier validation establishes the surrogate's experimental accuracy only transitively. As a direct check, a surrogate was trained for the parametric pier-cap geometry of Figure 7 with the method of Section 3, treating it as a further archetype in the data-generation sweep; it reached a coefficient of determination of 0.97 on held-out synthetic designs. The trained surrogate was then evaluated on the five experimental specimens. Table 4 and Figure 8 compare the measured ultimate load with the reference-solver and surrogate predictions.

The surrogate predictions track the reference solver closely and lie a mean factor of 1.44 below the measured loads, against 1.39 for the solver. The surrogate reproduces the solver, including its conservative bias for strut-crushing-governed D-regions, rather than introducing error of its own. This is the expected behaviour of a surrogate trained to reproduce the solver, and it confirms that the two tiers are consistent: the surrogate's experimental accuracy is, to within a few per cent, that of the reference solver it replaces.

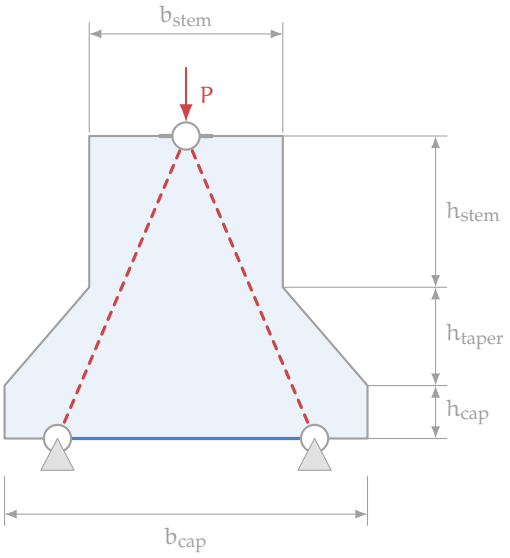


Figure 7: The parametric pier-cap model used for the direct validation of Section 4.4. The stepped D-region, comprising a cap band, a tapered transition and a stem, is described by the cap and stem widths b_{cap} and b_{stem}, the band heights h_{cap}, h_{taper} and h_{stem}, the out-of-plane thickness, the loading-plate width and the support layout. A concentrated load P on the stem spreads through inclined concrete struts to the supports, tied along the base. The Geevar and Menon specimens are one instance of this family.

Table 4: Direct validation of the surrogate against the five pier-cap experiments. Loads are total applied loads in kN.

Specimen	P_{exp}	solver	surrogate	P_{exp}/P_{surr}
S1	2224	1890	1681	1.32
S2	3068	1994	1948	1.57
S3	3436	2233	2166	1.59
S4	3608	2526	2430	1.48
S5	3464	2771	2780	1.25
mean				1.44

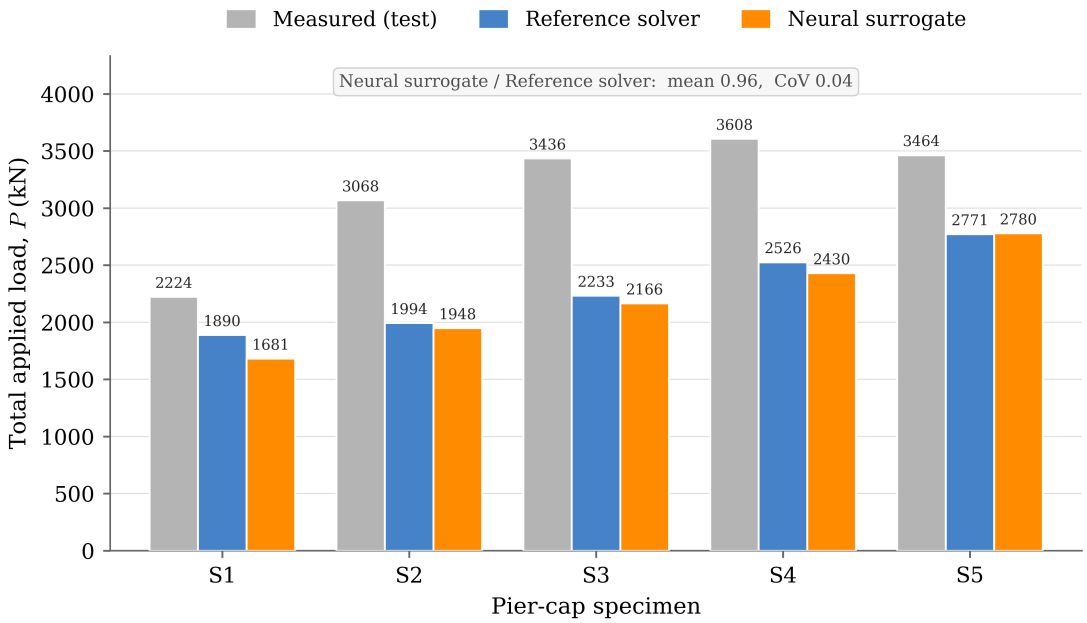


Figure 8: Direct experimental validation: the measured ultimate load of the five Geevar and Menon [4] pier-cap specimens, the CSFM reference-solver prediction, and the neural surrogate. The surrogate reproduces the solver and therefore inherits its conservative bias.

5 Results

The surrogate is evaluated on the held-out test split (designs the network never saw during training, Section 3.2), with the reference CSFM solver as the ground truth. Failure-load accuracy is reported as mean absolute percentage error (MAPE), root-mean-square error (RMSE) and coefficient of determination (R^2) against the reference CSFM solver failure load factor.

5.1 Failure-load prediction accuracy

Table 5 reports accuracy per archetype on the test split. A separate network is trained for each archetype. The metrics are computed over the designs that fail within the analysed load range; right-censored designs (which do not fail by $\lambda = 3.0$) are excluded from the per-archetype statistics and discussed below.

Table 5: Failure-load-factor prediction accuracy on the held-out test split, per archetype. n is the number of test designs that fail within the analysed load range.

Archetype	MAPE (%)	RMSE	R^2	n
Deep beam	5.5	0.117	0.965	92
Hammerhead	5.3	0.104	0.977	74
Multi-column bent	3.1	0.070	0.991	77
Pile cap	5.3	0.123	0.950	91

The surrogate reproduces the CSFM failure load factor with a coefficient of determination between 0.95 and 0.99 and a mean absolute percentage error of 3–5 % across all four D-region archetypes. The multi-column bent is predicted most accurately ($R^2 = 0.991$, MAPE 3.1 %) and the pile cap least accurately ($R^2 = 0.950$), though the spread across archetypes is small. Including the right-censored designs, for which the surrogate correctly predicts a load factor at or above the analysed ceiling, raises R^2 to 0.97–1.00, since those designs are well separated from the failing ones. Figure 9 shows the predicted against the reference CSFM solver failure load

factor for every test design; the points cluster tightly on the line of perfect agreement across the whole load-factor range.

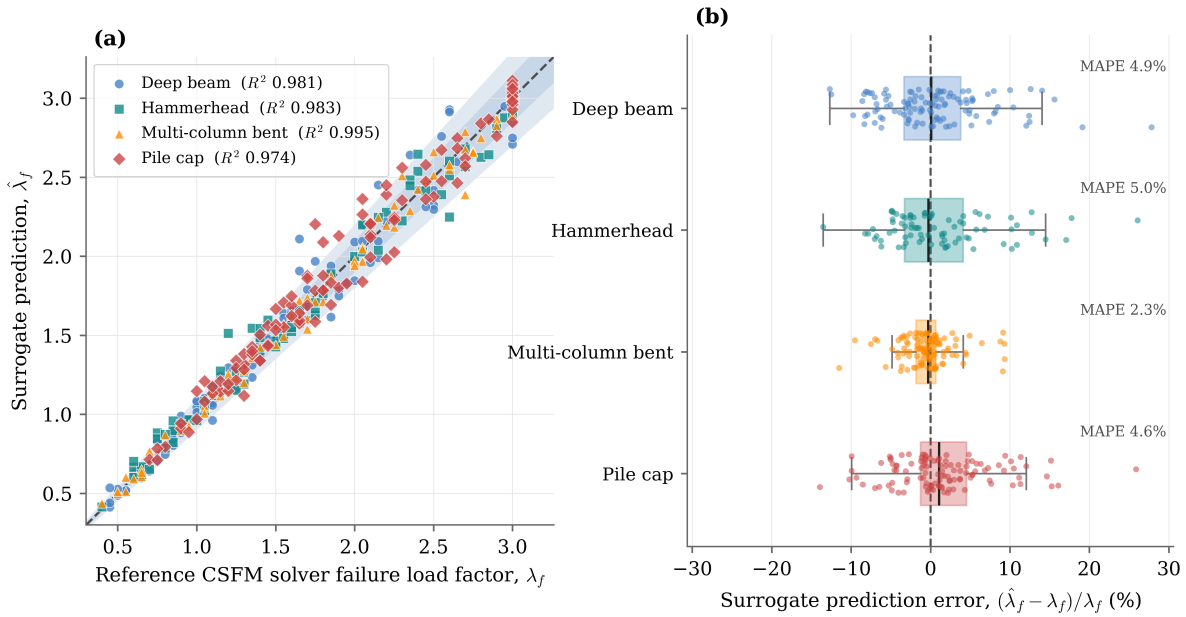


Figure 9: Surrogate accuracy on the held-out test split, for all four archetypes. (a) Predicted failure load factor $\hat{\lambda}_f$ against the reference Compatible Stress Field Method solver λ_f; the dashed line is perfect agreement and the shaded bands mark the $\pm 5\%$ and $\pm 10\%$ tolerances. (b) Distribution of the signed prediction error per archetype, with the box spanning the interquartile range and the mean absolute percentage error annotated. Almost every prediction falls inside the $\pm 10\%$ band. Metrics shown here are computed over all test designs, including the right-censored ones at $\lambda_f = 3.0$; Table 5 reports the stricter statistics restricted to genuine-failure designs.

5.2 Computational efficiency

Because the trained network predicts the failure load factor in a single forward pass, inference is far cheaper than the reference solver. On a single CPU core the surrogate evaluates a design in approximately 3.2 µs (batched), against approximately 4.1 ms per design for the reference CSFM analysis, a speed-up of roughly three orders of magnitude ($\approx 1300\times$). This is what makes the surrogate suitable for the design-space exploration, optimisation and reliability workflows that motivate this study.

5.3 Comparison with a baseline

To place the accuracy in context, a gradient-boosted-tree regressor [3] was trained on the same design parameters, the same per-archetype train/test split and the same target (the standard tabular-regression baseline). Table 6 compares the two on the genuine-failure test designs.

The neural surrogate is substantially more accurate than the gradient-boosted-tree baseline on every archetype: a coefficient of determination of 0.95–0.99 against 0.77–0.88, and a mean absolute percentage error of 3–5 % against 13–20 %. The failure load factor of a D-region is a smoothly varying but strongly coupled function of its design parameters, which the multilayer perceptron captures more faithfully than an axis-aligned tree ensemble.

Table 6: Failure-load accuracy of the neural surrogate against a gradient-boosted-tree baseline, per archetype (genuine-failure test designs).

	Gradient-boosted trees		Neural surrogate	
Archetype	MAPE (%)	R^2	MAPE (%)	R^2
Deep beam	20.1	0.773	5.5	0.965
Hammerhead	16.6	0.838	5.3	0.977
Multi-column bent	13.3	0.883	3.1	0.991
Pile cap	14.2	0.768	5.3	0.950

5.4 Predictive uncertainty

A point prediction alone is of limited use in the reliability and optimisation settings that motivate the surrogate: those workflows need a sense of how far each prediction can be trusted. Two standard tools are combined to provide this.

The first is a *bagged* deep ensemble [8]. For each archetype, ten networks are trained, each on an independent bootstrap resample of the training split, so that the members genuinely disagree where the training data is sparse rather than merely where their initial weights happen to differ. The ensemble mean $\bar{\lambda}_f$ is taken as the prediction and the across-member standard deviation σ as a raw uncertainty estimate. The ensemble mean matches the accuracy of the single network, with a coefficient of determination of 0.95–0.99 across the four archetypes, and σ is positively correlated with the actual prediction error (pooled Spearman rank correlation $\rho = 0.45$), so it does carry information about which predictions are least reliable. This correlation is moderate rather than strong, and σ on its own is not a precise predictor of the error of any single design; it does not need to be. The conformal step below requires only that σ rank design difficulty well enough for the interval width to track it, and a correlation of this strength is sufficient for that, while the coverage guarantee itself holds regardless of how well σ is correlated with the error.

Read directly, however, σ is *under-dispersed*: the members agree more than their error warrants, so a Gaussian interval $\bar{\lambda}_f \pm z\sigma$ covers far fewer test designs than its nominal level (the lower curve of Figure 10(b)). The second tool corrects this. Split-conformal calibration [1, 9] forms the σ-normalised nonconformity score $|\lambda_f - \bar{\lambda}_f|/\sigma$, evaluates it on the held-out validation split, and uses its empirical quantile q to construct the interval $\bar{\lambda}_f \pm q\sigma$. Because the validation designs are exchangeable with the test designs and were not used to fit the ensemble, this interval carries a finite-sample coverage guarantee, while its width still scales with σ and so widens for the designs the ensemble finds hard.

The calibration is effective. At a target coverage of 90 % the conformal multiplier is $q \approx 2.1$–2.6, and the resulting intervals cover 88–95 % of the test designs per archetype, 91 % pooled, in line with the target, with a mean half-width of about 0.10–0.21 in load-factor units. Figure 10(a) shows the ensemble prediction with its 90 % conformal interval, and Figure 10(b) confirms that the conformal coverage tracks the target across the whole range, where the raw ensemble spread does not. The surrogate therefore returns not a bare number but a calibrated, design-specific interval, which is the form of uncertainty a reliability analysis can use directly.

6 Discussion

The surrogate predicts the CSFM failure load factor to within 3–5 % across the four archetypes, substantially more accurately than the gradient-boosted-tree baseline, and at roughly three orders of magnitude lower cost than the reference solver.

The margin over the gradient-boosted-tree baseline is itself informative. The failure load factor of a D-region is a smooth but strongly coupled function of its geometry and material parameters: a change in one dimension shifts the inclination of every strut at once. A tree

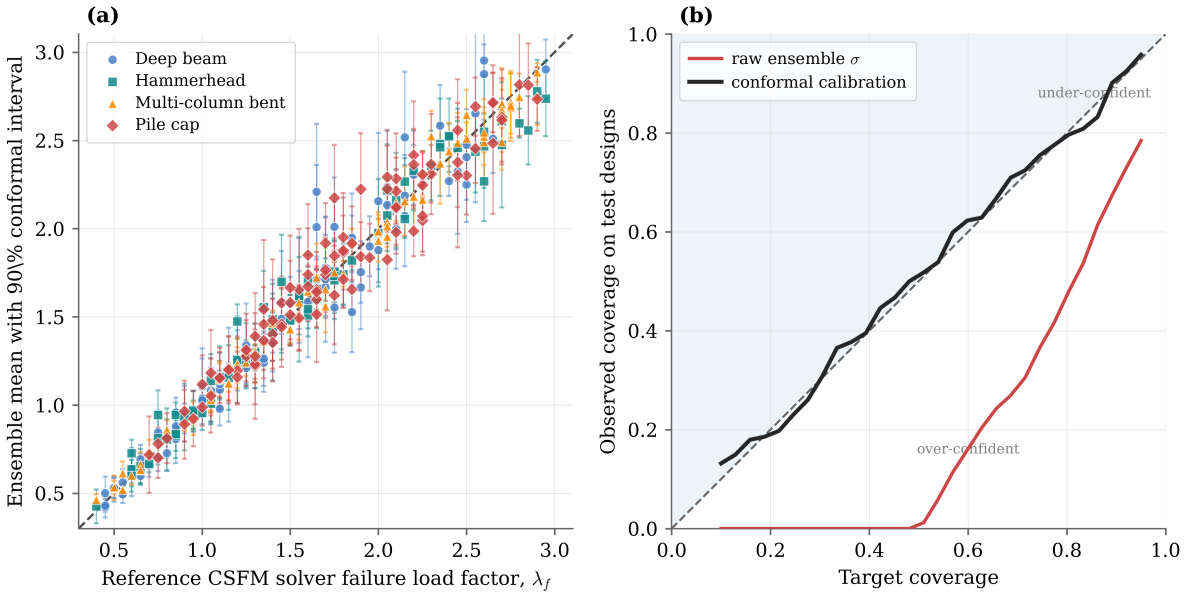


Figure 10: Predictive uncertainty of the surrogate, on the genuine-failure test designs of all four archetypes. (a) Bagged-ensemble mean prediction against the reference Compatible Stress Field Method solver, with bars showing the 90% split-conformal prediction interval; its width adapts to the ensemble spread. (b) Reliability diagram: the raw ensemble standard deviation, read as a Gaussian interval, is under-dispersed and falls below the diagonal, whereas the σ-normalised conformal interval tracks the target coverage, as its finite-sample guarantee requires.

ensemble approximates such a function with axis-aligned splits and piecewise-constant cells, which is inefficient when the response turns on combinations of inputs rather than on any single input; the multilayer perceptron, by contrast, mixes the inputs at its first layer and represents smooth interactions naturally. The gap is therefore one of inductive bias rather than of tuning, and it indicates that neural surrogates are the appropriate tool wherever the underlying mechanical response is smooth and coupled.

The surrogate inherits the conservative bias of the reference solver, which under-predicts the strength of strut-crushing-governed D-regions by a mean factor of about 1.4 against the pier-cap experiments (Section 4). For the design-screening use that motivates the surrogate this bias falls on the safe side and is acceptable: a fast, conservative estimate is well suited to ranking and filtering candidate designs. It does mean, however, that the surrogate predicts the capacity the reference solver would compute, not the true physical capacity; the two tiers of validation must be read together, and a final design check should still rest on the underlying method.

The bagged ensemble, with conformal calibration on top, attaches a calibrated prediction interval to each output rather than a bare number (Section 5.4). The interval widens for the designs the ensemble finds hard and its coverage is guaranteed in finite samples, so a large design sweep can be triaged automatically: the narrow-interval predictions are trusted and only the wide-interval ones referred back to the solver. This is the form of uncertainty a reliability analysis can act on, and obtaining it needed no change to the network, only a held-out calibration sample.

The accuracy is consistent across archetypes, with the pile cap the least accurate ($R^2 = 0.950$) and the multi-column bent the most ($R^2 = 0.991$). Because the strut and tie topology is fixed within an archetype, one network is trained per archetype; a single network spanning all archetypes would require a graph-based architecture that generalises across topologies, which is beyond the scope of the present discrete formulation. With that scope understood, the speed-up makes the surrogate directly usable in the design-space exploration, reinforcement optimisation and reliability workflows that motivate this study, with the caveat that its accuracy ceiling is

that of the reference solver validated in Section 4.

7 Limitations and future work

The method has four principal limitations. First, the surrogate inherits the strut and tie topology supplied by the host solver; it does not itself select or optimise that topology, and because the topology is fixed within an archetype, one network is trained per archetype; a single network spanning all archetypes would require a graph-based architecture. Second, the surrogate's accuracy ceiling is that of the reference solver, which is itself conservatively biased for strut-crushing-governed D-regions (Section 4). Third, the scope is restricted to the four archetypes of Section 3.2. Fourth, the experimental tier of the validation rests on a single five-specimen pier-cap series; it checks the reference solver for one D-region type but is not a broad experimental validation, and a wider benchmark spanning more geometries and failure modes would strengthen it. The conformal intervals of Section 5.4 are a further point to note: their coverage guarantee is *marginal*, holding on average over designs rather than conditionally for every design, so a sharper design-conditional guarantee remains open.

Addressing these limitations, in particular the conservative bias of the discrete formulation, the per-archetype scope and the design-conditional sharpening of the uncertainty intervals, is the focus of the studies that follow this one in a planned series, where a continuum treatment of the stress field is taken up in turn.

8 Conclusions

This study presented a neural-network surrogate for the failure load of concrete D-regions designed by the Compatible Stress Field Method. The network maps a D-region's design parameters directly to its failure load factor and its failure-state strut and tie member forces. Trained per archetype against a reference CSFM solver, the surrogate predicts the failure load factor with a coefficient of determination of 0.95–0.99 and a mean absolute percentage error of 3–5 % across four D-region archetypes, well above a gradient-boosted-tree baseline, and at roughly three orders of magnitude lower cost than the solver. The reference solver was itself checked against the experimental pier-cap benchmark of Section 4, so the surrogate is verified against the solver and the solver against measured strengths; a bagged deep ensemble with split-conformal calibration further attaches a calibrated prediction interval to each output. This study opens a planned series in which the limitations identified above are progressively addressed.

Data and code availability

The dataset-generation pipeline, the generated dataset, the trained model weights and the evaluation code will be released in a public repository and archived with a permanent identifier on publication.

Declaration of competing interest

The authors declare no competing financial interests or personal relationships that could have influenced the work reported in this study.

References

[1] Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

[2] CEN. *EN 1992-1-1:2004 Eurocode 2: Design of Concrete Structures – Part 1-1: General Rules and Rules for Buildings*. European Committee for Standardization, Brussels, 2004.

[3] 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*, pages 785–794, 2016.

[4] Indri I. Geevar and Devdas Menon. Strength of reinforced concrete pier caps — experimental validation of strut-and-tie method. *ACI Structural Journal*, 116(1):261–272, 2019. doi: 10.14359/51711138.

[5] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[6] Walter Kaufmann, Jaime Mata-Falcón, Manuel Weber, and Tena Galkovski. *Compatible Stress Field Design of Structural Concrete: Principles and Validation*. ETH Zurich and IDEA StatiCa s.r.o., Zurich, 2020. ISBN 978-3-906916-95-8.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

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

[9] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018.

[10] Peter Marti. Basic tools of reinforced concrete beam design. *ACI Journal*, 82(1):46–56, 1985.

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

[12] Aurelio Muttoni, Joseph Schwartz, and Bruno Thurlimann. *Design of Concrete Structures with Stress Fields*. Birkhauser, Basel, 1997.

[13] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[14] Esteban Samaniego, Cosmin Anitescu, Somdatta Goswami, Vien Minh Nguyen-Thanh, Hongwei Guo, Khader Hamdia, Xiaoying Zhuang, and Timon Rabczuk. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 362:112790, 2020.

[15] Jorg Schlaich, Kurt Schafer, and Mattias Jennewein. Toward a consistent design of structural concrete. *PCI Journal*, 32(3):74–150, 1987.

[16] Frank J. Vecchio and Michael P. Collins. The modified compression-field theory for reinforced concrete elements subjected to shear. *ACI Journal*, 83(2):219–231, 1986.

[17] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43 (5):A3055–A3081, 2021.