

Article

A DNN-Based Surrogate Constitutive Equation for Geometrically Exact Thin-Walled Rod Members

Marcos Pires Kassab ^{1,*} , Eduardo de Moraes Barreto Campello ¹  and Adnan Ibrahimbegovic ² 

¹ Department of Structural and Geotechnical Engineering, Polytechnic School, University of São Paulo, P.O. Box 61548, São Paulo 05424-970, SP, Brazil; campello@usp.br

² Laboratoire Roberval, Centre de Recherche Royallieu, University of Technology Compiègne—Alliance Sorbonne University, Rue de Docteur Schweitzer, Hauts-de-France, 60200 Compiègne, France; adnan.ibrahimbegovic@utc.fr

* Correspondence: marcos.kassab@usp.br

Abstract: Kinematically exact rod models were a major breakthrough to evaluate complex frame structures undergoing large displacements and the associated buckling modes. However, they are limited to the analysis of global effects, since the underlying kinematical assumptions typically take into account only cross-sectional rigid-body motion and occasionally torsional warping. For thin-walled members, local effects can be notably important in the overall behavior of the rod. In the present work, high-fidelity simulations using elastic 3D-solid finite elements are employed to provide input data to train a Deep Neural Network (DNN) to act as a surrogate model of the rod's constitutive equation. It is capable of indirectly representing local effects such as web/flange bending and buckling at a stress-resultant level, yet using only usual rod degrees of freedom as inputs, given that it is trained to predict the internal energy as a function of generalized rod strains. A series of theoretical constraints for the surrogate model is elaborated, and a practical case is studied, from data generation to the DNN training. The outcome is a successfully trained model for a particular choice of cross-section and elastic material, that is ready to be employed in a full rod/frame simulation.

Keywords: geometrically exact rod; surrogate models; Deep Neural Network; Machine Learning



Academic Editor: Demos T. Tsahalis

Received: 12 December 2024

Revised: 31 January 2025

Accepted: 5 February 2025

Published: 3 March 2025

Citation: Kassab, M.P.; Campello, E.d.M.B.; Ibrahimbegovic, A. A DNN-Based Surrogate Constitutive Equation for Geometrically Exact Thin-Walled Rod Members.

Computation **2025**, *13*, 63. <https://doi.org/10.3390/computation13030063>

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license

(<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Kinematically exact rod models have been studied for many decades. Beginning from 2D model from Reissner [1], the subject only matured when finite rotation parametrization was consistently derived, with works such as Argyris [2] and Ibrahimbegovic's [3]. From then on, the 3D formulation was conceived: works from Simo [4,5] can be regarded as the pioneer ones with fully 3D description, despite inconsistent linearization of the weak form. Following contributions improved certain aspects of Simo's initial formulation. Classical works such as Pimenta and Yoho [6] and Ibrahimbegovic [7] already featured consistent linearization. Up to those works, only Timoshenko-like assumptions had been considered: shear flexible cross-sections that performed only rigid body motion in the space. Such assumptions are not suitable for problems featuring torsion. For this reason, efforts were made to introduce additional warping degree of freedom (DOF), inspired by classical Vlasov's work [8], see, e.g., Vu Quoc and Simo [9], Campello and Pimenta [10,11], Gruttmann [12] and Gonçalves [13]. Another limitation was that only truncated constitutive equations (or strain measures, which renders the same effect) were available, preventing certain global buckling modes from being detected. This motivated works from, Campello

and Lago [14] and Kassab et al. [15], which for the first time employed the complete constitutive equation for Saint-Venant's and Simo-Ciarlet's hyperelastic materials, solving issues related to critical load detection and post-critical path development in pathological cases. Other works focused on introducing inelastic behavior in the geometrically exact thin-walled rods, e.g., [12] and Kassab et al. [16,17]. Some authors also tried to introduce in the formulation extra DOFs to represent cross-sectional distortion. Multiple approaches are possible, such as adding linear combination of distortion modes (e.g., Pimenta and Campello [18], Dasambiagio [19] and Gonçalves [20]).

It is the spirit of this work to be able to capture in a lower-order representation local effects, an impossible task if one attains to the classical stress integration constrained to the rods kinematics. Automating the process of finding constitutive equation is a state-of-art subject. In recent works, Lorenzis and collaborators [21–23] have been using techniques that relies on assembling a library of building blocks for constitutive equations, specially chosen to satisfy desirable mechanical properties, such as objectivity and polyconvexity. Even though this approach works remarkably well for finding constitutive models with minimal amount of parameters for 3D solid elements, it is not suitable for operating in terms of generalized stress resultants/generalized strains for rods. Since the quantities of interest are not stresses themselves, but rather its integrals over the cross-section, the usual building blocks and mechanical requirements might not suit the present needs. For instance, it is not reasonable to require polyconvexity at this level. Also, since there is direct dependence on the cross-section geometry and the mechanical behavior of the structure, it is not trivial to define an appropriate sub-base of the function space to undergo the optimization problem.

DNNs are powerful tools to solve inverse problems. Different engineering applications have benefited from this approach, not only in solid mechanics. They excel in tasks that traditionally would require multiple evaluations of expensive high-fidelity simulations, such as reliability analysis [24] and topological optimization [25–27]. It has also been largely used for model order reduction in scenarios that a restricted amount of control parameters are used to predict global behavior across an extended domain [28], to extract homogenized properties [29,30]. By adding information about the underlying physics, less amount of training data is needed [31,32]. Also, complex tasks such as mesh discretization and multiscale analysis are possible [33]. Real-time control problems also benefit from data-based approaches, such as in [34]. For the aforementioned reasons, the choice was to construct a Deep Neural Network-based (DNN-based) surrogate constitutive equation. By using generalized rod strains and internal energy evolution from high-fidelity 3D-solid finite element simulations, a framework that relies solely on synthetic data was formulated.

The approach is designed to indirectly take into account local effects such as web/flange bending and buckling, while retaining only the usual rod DOFs. This feature follows from data gathered from refined (high-fidelity) simulations and, together with the DNN-based framework itself, stands as one of the novel aspects of this work. Physical requirements are met by construction, and automatic differentiation tools shall allow direct introduction of the surrogate model in already existing frameworks. To employ Machine Learning techniques in frame structures might not look particularly new. Many works tackling different aspects of real-life challenges are available, such as usual I-section beams [35,36], sections with openings [37–40], corrugated beams [41], reinforced concrete beams [42–44] and even for more complex structural systems [45]. All those works have their own merits, but we highlight that they focus on finding ultimate states (or load bearing capacity), overlooking the stress-strain constitutive equation and state evolution during progressive loading. This gives our work an innovative status.

Only the bi-dimensional rod context is of interest for now, even though the input solid models are three-dimensional. In this procedure, the biggest amount of computational effort is data generation.

This is a work in progress, and the current scope is to provide:

- description of the theoretical requirements of the modelled energy function;
- practical aspects about the NN implementation;
- illustrative example.

In the present work, the surrogate model that is actually trained is restricted to the 2D frame case for a specific geometry (cross-section and member length). We envision on setting the stage for developing more robust surrogate models in future works. Those are intended to be used within already existent rod FEM frameworks. On a future work, full rod model implementation will take place. By using automatic differentiation tools (native from many ML environments), one shall be able to calculate first and second derivatives w.r.t strains to recover stress resultants and material tangent stiffness, respectively. In the present work, Pytorch v2.4.0 [46] is used to implement the DNN.

The organization of the paper is as follows: in Section 2, the basic description of the rod model is displayed. In Section 3, minimal requirements of the surrogate model and implementational aspects are described. In Section 4, a practical case is studied, and different DNN architectures are explored. Finally, in Section 5 we close the paper with our conclusions.

Throughout the text, the notation is as follows (except where clearly stated otherwise): Greek and Latin italic lowercase letters are scalar quantities ($a, b, \alpha, \beta, \dots$), bold Greek and Latin italic lowercase represent vectors ($\mathbf{a}, \mathbf{b}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \dots$) and bold Greek and Latin italic uppercase denote second order tensors in three-dimensional Euclidean space ($\mathbf{A}, \boldsymbol{\Omega}, \dots$). Summation convention over repeated indices is adopted (Einstein's notation), with Greek letter indices ranging from 1 to 2 and Latin indices from 1 to 3. Derivative with respect to the rod's axial coordinate is represented by $(\circ)'$ and partial derivatives by either $\frac{\partial(\circ)}{\partial(\star)}$ or $(\circ)_{(\star)}$.

2. A Brief Description of the Base Rod Model

The basic rod model of interest is the one described in Campello and Pimenta [10], Campello and Lago [14] and Kassab et al. [15]. General considerations about nonlinear solids mechanics and its application in structures can be found in works such as [47,48]. For the sake of completeness and for compatibility with our previous work, the full 3D formulation is shown along this section. However, this work is restricted to the 2D case, in which only some of the degrees of freedom of the full 3D description are needed. Appropriate remarks are provided where needed to reduce the complete formulation to the desired 2D context.

The kinematics of the rod model are summarized in Figure 1.

For a rod with reference length L^r and cross-sectional area A^r , it is defined a straight initial configuration, taken as the reference one. In the reference orthonormal base $\{\mathbf{e}_1^r, \mathbf{e}_2^r, \mathbf{e}_3^r\}$, the reference axial direction is $\mathbf{e}_3^r$, whereas $\{\mathbf{e}_1^r, \mathbf{e}_2^r\}$ defines the cross-sectional reference plane. Thus, points of the reference domain are defined by the vector field

$$\boldsymbol{\zeta} = \mathbf{a}^r + \boldsymbol{\zeta}, \quad (1)$$

where $\mathbf{a}^r = \zeta_\alpha \mathbf{e}_\alpha^r$, $\{\zeta_1, \zeta_2\} \in A^r$ is the director vector that maps the cross-sectional points w.r.t to the cross-sectional origin and $\boldsymbol{\zeta} = \zeta \mathbf{e}_3^r$, $\zeta \in \Omega^r = [0, L^r]$ is the axial reference position, both in the reference configuration. The current configuration is obtained by the

composition of a cross-sectional rigid body motion (axial displacement $\mathbf{u}$ and cross-sectional rotation tensor $\mathbf{Q}$) and a warping $\mathbf{w}$, rendering

$$\mathbf{x} = \mathbf{z} + \mathbf{y}, \quad (2)$$

with

$$\mathbf{z} = \boldsymbol{\zeta} + \mathbf{u}, \quad \mathbf{y} := \mathbf{a} + \mathbf{w}, \quad \mathbf{a} = \mathbf{Q}\mathbf{a}^r, \quad \mathbf{w} := p\psi\mathbf{e}_3. \quad (3)$$

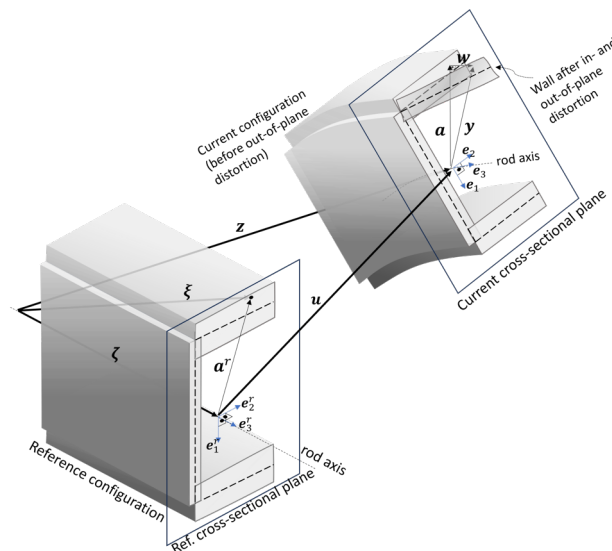


Figure 1. Description of kinematics for the rod model.

The rotation tensor $\mathbf{Q}$ is parametrized by means of Euler-Rodrigues formula

$$\mathbf{Q} = \mathbf{I} + \frac{\sin \theta}{\theta} \boldsymbol{\Theta} + \frac{1 - \cos \theta}{\theta^2} \boldsymbol{\Theta}^2, \quad (4)$$

in which $\boldsymbol{\theta}$ is the rotation vector with magnitude $\theta = \|\boldsymbol{\theta}\|$ and $\boldsymbol{\Theta} = \text{skew}(\boldsymbol{\theta})$. For the warping displacement $\mathbf{w}$, p is the magnitude associated to the warping shape function ψ . The relevant kinematical quantities are now collected in the generalized displacement vector $\mathbf{d} = [\mathbf{u}, \boldsymbol{\theta}, p]^T$.

Following the same development as in the previous works, one can define the back-rotated deformation gradient tensor $\mathbf{F}^r = \mathbf{Q}^T \mathbf{F}$

$$\mathbf{F}^r := \mathbf{I} + \gamma_\alpha^r \otimes \mathbf{e}_\alpha^r + \gamma_3^r \otimes \mathbf{e}_3^r, \quad (5)$$

with

$$\gamma_\alpha^r = \psi_{,\alpha} p \mathbf{e}_3^r, \quad \gamma_3^r = \boldsymbol{\eta}^r + \boldsymbol{\kappa}^r \times \mathbf{y}^r + \psi p' \mathbf{e}_3^r, \quad (6)$$

and $\boldsymbol{\eta}^r := \mathbf{Q}^T \mathbf{z}' - \mathbf{e}_3^r$, $\boldsymbol{\kappa}^r := \mathbf{Q}^T \boldsymbol{\kappa}$, $\boldsymbol{\kappa} := \text{axial}(\mathbf{Q}' \mathbf{Q}^T)$. Now, the generalized strain vector is defined as $\boldsymbol{\varepsilon}^r := [\boldsymbol{\eta}^r, \boldsymbol{\kappa}^r, p, p']^T$.

The weak form of the equilibrium is given by (see [10,15] for details)

$$\delta W = \delta W_{int} - \delta W_{ext} = \int_{\Omega^r} \boldsymbol{\sigma}^r \cdot \delta \boldsymbol{\varepsilon}^r d\zeta - \int_{\Omega^r} \hat{\mathbf{f}} \cdot \delta \mathbf{d} d\zeta = 0, \quad \forall \delta \mathbf{d}, \quad (7)$$

where σ^r is the vector of (back-rotated) cross-sectional stress-resultants, given by

$$\sigma^r := \begin{bmatrix} n^r \\ m^r \\ Q \\ B \end{bmatrix} = \int_{A^r} \begin{bmatrix} \tau_3^r \\ (a^r + p\psi e_3^r) \times \tau_3^r \\ \psi_{,\alpha} \tau_\alpha^r \cdot e_3^r + \psi \tau_3^r \cdot (\kappa^r \times e_3^r) \\ \psi \tau_3^r \cdot e_3^r \end{bmatrix} dA^r, \quad (8)$$

in which n^r is the vector with three components of back-rotated shear and axial forces, m^r is the vector that holds three components of back-rotated bending and torsional moments, Q is the bi-shear and B is the bi-moment, for the columns τ_i^r of the back-rotated first Piola-Kirchoff stress tensor $P^r = Q^T P = \tau_i^r \otimes e_i^r$. Also, $\hat{f} := [\hat{n} \ \hat{\mu} \ \hat{B}]^T$ is the vector of applied external resultants, with external forces $\hat{n}$, external pseudo-moments $\hat{\mu}$ and external bi-moment $\hat{B}$ as defined in [15]. In the mentioned bibliography, one finds also definitions of the tangent stiffness operator, necessary for numerically solving Equation (7) through Newton-Rapson iterations. This operator has three terms: material, geometric and external loading. Here, it is only of interest to mention the material tangent stiffness D .

In order to effectively compute σ^r and D , a constitutive equation must be chosen. Here, we assume that the material is hyperelastic, thus a specific energy function $\Psi^{el} = \hat{\Psi}^{el}(\epsilon^r)$ exists, wherefrom $\sigma^r := \frac{\partial \Psi^{el}}{\partial \epsilon^r}$. In the current context, Ψ^{el} itself is not known beforehand and the goal of this work is to employ a DNN to generate a surrogate model for it. Moreover, it is the spirit of this work to be able to represent local effects (flange/web bending and buckling) by means of such reduced order surrogate model. Even though only global DOFs are present, the constitutive equation shall take into account such effects from appropriate training of the surrogate model. Once an expression for Ψ^{el} is established, automatic differentiation tools can be used to evaluate

$$\sigma^r := \frac{\partial \Psi^{el}}{\partial \epsilon^r} \quad \text{and} \quad D := \frac{\partial \sigma^r}{\partial \epsilon^r} = \frac{\partial^2 \Psi^{el}}{\partial \epsilon^r \partial \epsilon^r}. \quad (9)$$

We recall that in the context of geometrically-exact rod theories, the usual approach is to take a continuum mechanics material behavior through an energy function Ψ^{el} , compute the stresses via differentiation of the energy w.r.t. the strains (different stress-strain conjugate pairs maybe used), and then, and then directly integrate the stresses as in Equation (8), for obtaining the cross-sectional resultants. Ultimately, the stress resultants σ^r are function of ϵ^r and of the cross-section geometry. This standard procedure is followed in many established formulations, and is thoroughly demonstrated in our previous work [15], where we also show the importance of using un-truncated forms of the resulting constitutive equation for correctly capturing the needed strain coupling for certain buckling problems. Usually, this integration does not have a closed form solution (numerical integration is usually required) and the approach is limited by the rod assumptions. Here, the approach is different: we abandon the direct integration of stresses and write the stress resultants as derivatives of the internal energy function, respecting the definition of hyperelastic materials. By doing this, one can avoid limitations that arise from direct imposition of rod assumptions, provided that enough data is available to fine-tune the energy function. Here, data is gathered from refined simulations with 3D solid finite elements.

For the current work, only two-dimensional rod problems are of interest, as in a Timoshenko rod. Therefore, from now on, it is assumed that $u_1 = \theta_2 = \theta_3 = p = 0$, $\eta_1^r = \kappa_3^r = p' = 0$, $n_1^r = m_2^r = m_3^r = B = Q = 0$. From the rod DOFs, generalized strains and stresses, only $u_2, u_3, \theta_1, \eta_2^r, \eta_3^r, \kappa_1^r, n_2^r, n_3^r$ and m_1^r are non-zero. To simplify the notation, from now on the vectors d, ϵ^r and σ^r refers to the original ones but with the zero values suppressed.

3. Building and Training the Surrogate Model

A given prismatic rod member is characterized by its reference cross-sectional area A^r , length L^r and material properties (parameters and constitutive equation at stress-strain continuum level). Therefore, the constitutive equation that connects generalized rod strains (ϵ^r) and generalized rod stresses (σ^r) shall depend on those characteristics.

In the present work, we do not intend to create a surrogate model that is capable of dealing with any cross-section and material properties, as an immense amount of training data would be required. Instead, we choose to focus on a specific geometry (A^r and L^r) and with fixed material properties, see Section 4 for details. Of course, this limits the applicability of the proposed framework. Our current implementation should be regarded as a proof of concept of an innovative approach. However, for the chosen geometry, we shall allow a comprehensive set of boundary conditions when training our surrogate model.

One of the outcomes of the proposed approach is that it shall be able to detect local effects such as softening-like behavior as the local buckling progresses. As for illustration, one can see in Appendix B that at higher strain rates, the energy curves becomes straight, which indicates loss of stiffness and maximum load bearing capacity for the cross-section. To some extent, this phenomena looks like a softening that occurs in the inelastic context, such as in [49], but in the fully elastic domain. Differently from simpler multi-linear approaches (see [50]), that rely on defining simplified stress-strain diagrams, the generated model is actually nonlinear, and shall naturally take into account the intricate strain coupling that arises in finite strain scenario. Even though this global effect is recovered for the low-order surrogate model, information about the local buckling modes are lost. This is not an issue for us here, since local studies can always be performed as a post-processing step in a higher-order framework, whilst the computationally challenging global solution is achieved efficiently within low-order models.

To generate a surrogate model for specific elastic energy function, we choose a DNN-based approach. Using synthetic data from refined 3D-solid FEM simulation as inputs, we are going to explore two different architectures for the surrogate model:

- Direct estimation of the energy function $\Psi^{el} \approx \Psi_{(1)}^{surr} = \Psi^{NN}(\epsilon^r)$, where Ψ^{NN} is a $\mathbb{R}^8 \rightarrow \mathbb{R}$ function, made of densely connected linear layers and activation function from the linear unit family (ELU, ReLU, GELU, SILU, etc.), see Figure 2a;
- Estimation of a multiplicative function that ponderates the linear elastic solution, i.e., the quadratic function $\Psi^{el} \approx \Psi_{(2)}^{surr} = \chi^{NN} \Psi_{quad}$, where χ^{NN} is a $\mathbb{R}^8 \rightarrow \mathbb{R}$ function, made of densely connected linear layers and activation function from the linear unit family (ELU, ReLU, GELU, SILU, etc.) and a final logistic sigmoid activation layer and $\Psi_{quad} = \frac{1}{2} \epsilon^{rT} D \epsilon^r$, with the tangent stiffness for rods D (as in [11] or [15], for example), see Figure 2b.

In both cases, the DNNs will be trained for normalized input strain ($\tilde{\epsilon}^r$) and energy ($\tilde{\Psi}^{el}$) values, as it will be shown in Section 3.3. The exact network architecture (either Ψ^{NN} or χ^{NN}) should be studied for our specific case. For the current work, Pytorch [46] was used for implementation of the DNNs.

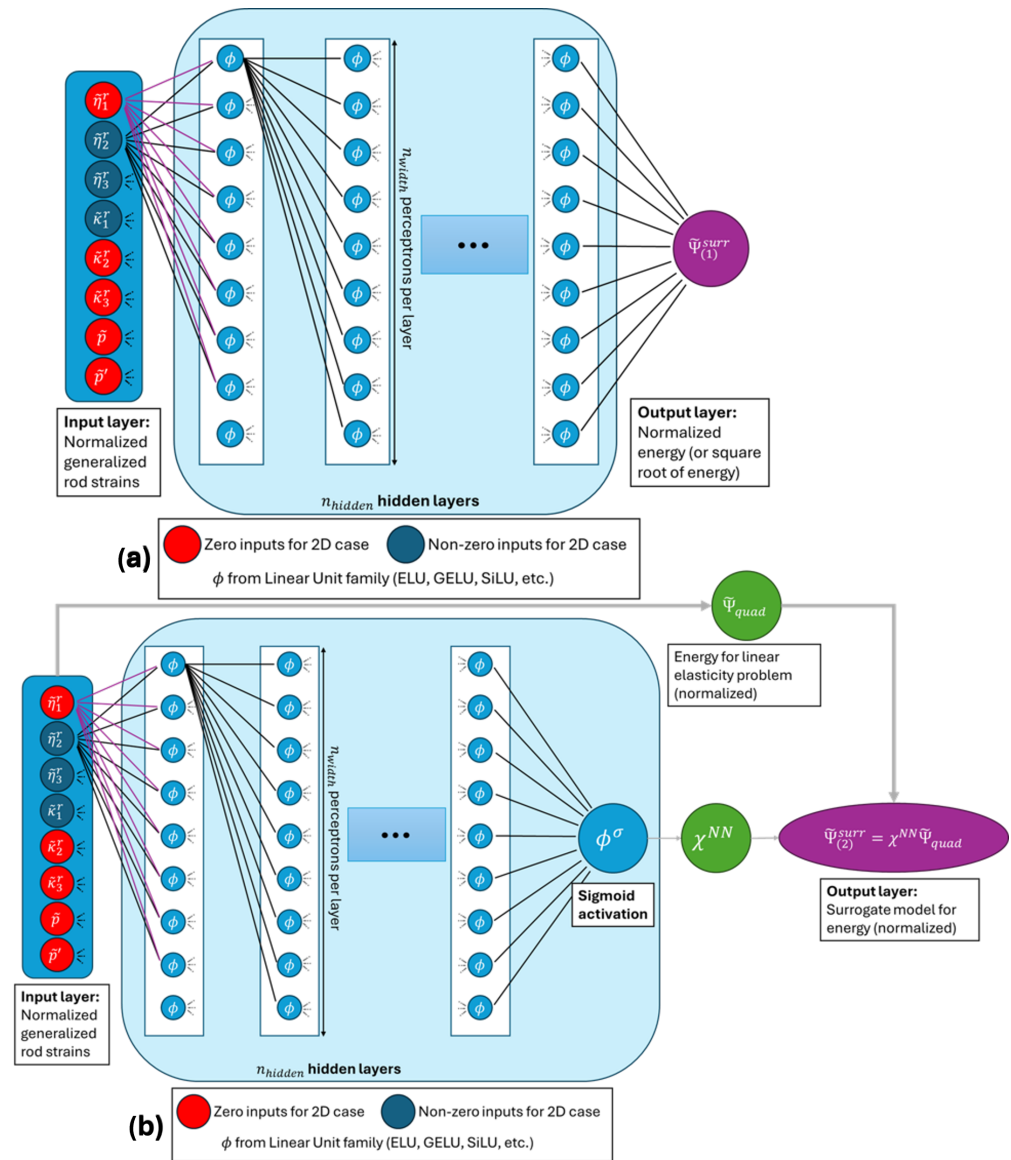


Figure 2. Proposed DNN architectures and properties for the current application, for (a) $\Psi_{(1)}^{surr}$ and (b) $\Psi_{(2)}^{surr}$.

3.1. Theoretical Considerations

The surrogate model must satisfy the following constraints:

1. Objectivity;
2. Zero strain energy for undeformed configurations: $\epsilon^r = \mathbf{o} \Rightarrow \Psi^{el} = 0$
3. Differentiability (to be able to obtain σ^r and D): the model must be twice differentiable, at least in a piecewise sense ($\Psi^{el} \in W^{2,\infty}$ in Sobolev space notation), and ideally $\Psi^{el} \in C^2$;
4. Non-negative energy: $\Psi^{el} \geq 0, \forall \epsilon^r$

For the two proposed architectures, different measures are required to guarantee compliance to those constraints. Condition 1 is automatically met since the argument ϵ^r is objective (see, e.g., [6,10,11]). For $\Psi_{(1)}^{surr}$, condition 2 is guaranteed if unbiased layers are used alongside activation functions satisfying $\phi(0) = 0$. For $\Psi_{(2)}^{surr}$, it is satisfied by construction, since $\Psi_{quad}(\mathbf{o}) = 0$. Condition 3 is satisfied by choosing sufficiently smooth ϕ . From conditions 1 and 2, we conclude that functions from the *linear unit* family are a suitable choice for the task. Condition 4 cannot be enforced directly for $\Psi_{(1)}^{surr}$, and shall be

learnt during training process and the final result must be verified. However, for $\Psi_{(2)}^{surr}$, it is satisfied also by construction, since $\chi^{NN}, \Psi_{quad} > 0$.

It is the spirit of the work to be able to represent in the low-level rod element in the Finite Element Method (FEM) framework. Thus, the internal energy and its derivatives in the integration points must be calculated. Since the authors intend to use this approach in usual 2-noded rod elements with linear interpolation, we may write for the shape functions

$$N_1(\iota) = \frac{1}{2}(-\iota + 1), \quad N_2(\iota) = \frac{1}{2}(\iota + 1), \quad (10)$$

for natural coordinate $\iota \in [-1, 1]$. With reduced integration using Gauss's method, the only integration point is positioned exactly at the axial mid-point ($\iota = 0$). Thus, to solve the equilibrium weak form (7) one need only to evaluate $\sigma^r|_{\iota=0}$ and $D|_{\iota=0}$. Conveniently, the total internal energy of a given rod element can be approximately evaluated with

$$W^{el} = \int_{\Omega^r} \Psi^{el} d\zeta \approx \sum_{i=1, n_g} w_i \Psi_i^{el} \stackrel{n_g=1}{=} \Psi^{el}|_{\iota=0} L^r. \quad (11)$$

Thus, for this work, the specific strain energy at the integration point in the rod element is calculated by

$$\Psi^{el}|_{\iota=0} = \frac{W^{el}}{L^r}. \quad (12)$$

3.2. Data Generation

The data needed for the surrogate model is generated from a high fidelity simulation (e.g., using FEM 3D solid elements). To be compatible with 2-noded rod elements, equivalent boundary conditions (BCs) are applied only at the edges (essentially, imposed some generalized displacement $\hat{d}$ at the ends for the 2D case). Since ε^r is objective, training process can be greatly simplified, given that there is no need to generate combinations of boundary conditions that make the final configurations differ only from rigid body motion. Hence, one can select one of the ends and clamp it ($\hat{d} = \mathbf{o}$), while varying the opposing end's generalized displacement, as in Figure 3. To apply the boundary conditions in the FEM solid models, one has to select all the nodes of the end cross section (in red in Figure 3), and apply a composition of displacements stemming from axial displacements $\bar{u}_2$ and $\bar{u}_3$ and from the flexural rotation $\bar{\theta}_1$. The latter yields both axial and transversal components, which vary with the rotation angle and increase linearly w.r.t to the distance to the axis. Within this approach, local elastic phenomena such as web/flange bending and buckling are naturally dealt with in the high fidelity simulation, and information about them are embedded in the total strain energy. Thus, one can indirectly correlate rod generalized strain to the impact of local phenomena in the cross-sectional stress resultants.

Afterwards, the element's total strain energy W^{el} is retrieved as an output of the high fidelity simulation (in our case, FEAP 3D-solid finite element simulation). This kind of structural simulation usually is performed using incremental loading procedures. Thus, for an input target value of $\bar{d}$, one gets a curve $\gamma \bar{d}$ vs. BW^{el} , with $0 \leq \gamma \leq 1$.

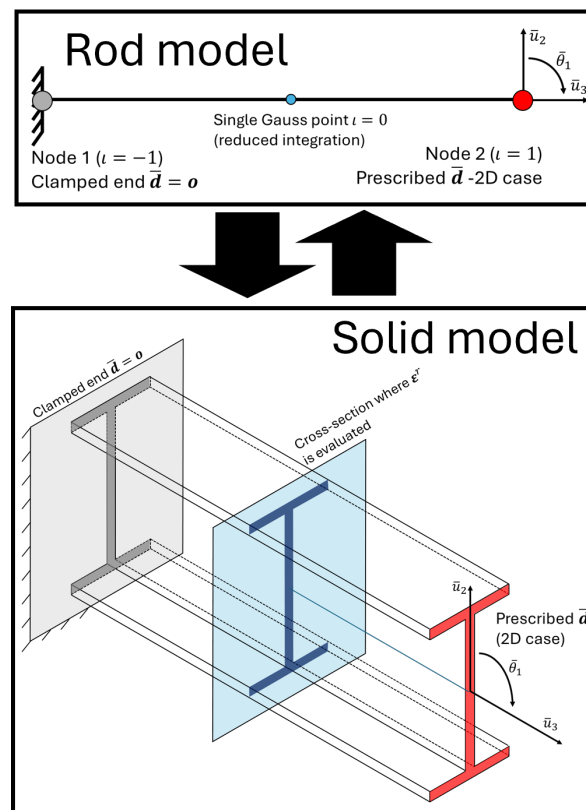


Figure 3. Generic boundary conditions for data generation using solid elements.

Using (10), for a given value of imposed displacement at one end $\hat{\mathbf{d}} = \gamma \bar{\mathbf{d}}$, one can evaluate the displacement $\mathbf{d}|_{l=0}$, the associated axial derivatives $\mathbf{d}'|_{l=0}$ and then the strain $\boldsymbol{\varepsilon}^r|_{l=0}$. Afterwards, one can calculate the specific energy $\Psi^{el}|_{l=0}$ using (12). The pairs $(\boldsymbol{\varepsilon}^r|_{l=0}, \Psi^{el}|_{l=0})$ are the required inputs for the DNN training process.

Training and testing sample sets must be generated either procedurally, randomly, or both, until the resulting network has attained a sufficient level of accuracy. In the current work, the adopted procedure for data generation consists in

1. setting upper and lower boundaries for each entry of $\hat{\mathbf{d}}$, creating a hypercubic domain for the study, see Figure 4;
2. procedurally generating solid simulation input data in the boundaries of the study domain (i.e., vertex, edges and faces of the domains' hypercube), equally spaced, for training purposes (see green dots in Figure 4);
3. randomly generating solid simulation input data in the faces of the domain's hypercube, for both training and testing (see purple dot in Figure 4).

Since data is synthetic, there is no theoretical limit to the amount of generated samples. It is suggested that the amount of testing data should be 15–25% of the training data.

In order to avoid influence of the scale of the input quantities $\hat{\mathbf{d}}$ in the sampling process, the random picking process should be split in two: first, select randomly one face of the hypercube (every face must have the same probability of being chosen), and then, one point of the face is randomly chosen (with uniform probability distribution). In Figure 4, one can see the data points generated from steps (1) and (2), alongside one randomly generated simulation from step (3).

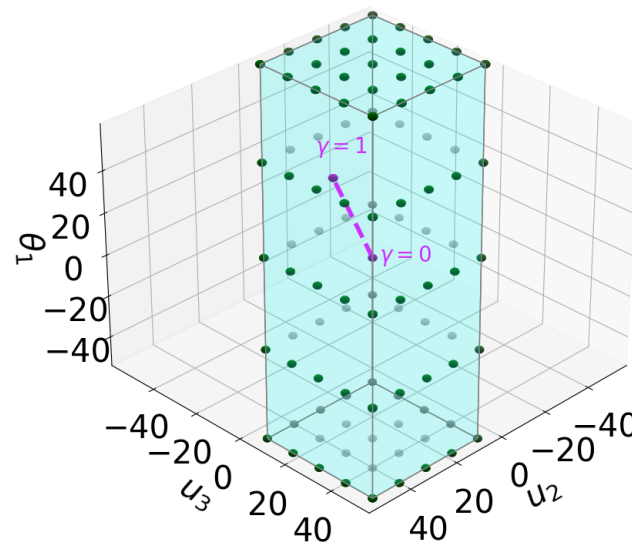


Figure 4. Regular grid—maximum input displacements (green dots) and one curve with random maximum displacement (purple dots/dashed line). Hypothetical case.

3.3. Training Process

The first step is to calculate the pairs (ϵ^r, Ψ^{el}) as specified in Section 3.2. Then, organize the generated data as a table with $n_{\text{input data}}$ lines $(\epsilon^{rT}, \Psi^{el})$. Its columns are given by

$$\begin{aligned} \epsilon_i^r &= \left[(\epsilon_i^r)^{(1)} \quad (\epsilon_i^r)^{(2)} \quad \dots \quad (\epsilon_i^r)^{(k)} \quad \dots \quad (\epsilon_i^r)^{(n_{\text{input data}})} \right]^T, \\ \Psi^{el} &= \left[(\Psi^{el})^{(1)} \quad (\Psi^{el})^{(2)} \quad \dots \quad (\Psi^{el})^{(k)} \quad \dots \quad (\Psi^{el})^{(n_{\text{input data}})} \right]^T, \end{aligned} \quad (13)$$

with

- ϵ_i^r , with i ranging from 1 to 8 for the full 3D case, and from 1 to 3 in the 2D case, representing each generalized rod strain component;
- Ψ^{el} a vector that collects all input data for energy in the refined model simulation.

Afterwards, training follows usual NN practices. Variables should be normalized to ensure better convergence behavior of the optimizer algorithm. The adopted normalization factor was the maximum absolute value for each component of ϵ^r and Ψ^{el} . Hence, one gets the normalized quantities ($\tilde{\circ}$)

$$\tilde{\epsilon}_i^r = \frac{\epsilon_i^r}{|\epsilon_i^r|_\infty}, \quad \tilde{\Psi}^{el} = \frac{\Psi^{el}}{|\Psi^{el}|_\infty}, \quad (14)$$

where $|\circ|_\infty := \max|\circ|$. From Equation (14), it is clear that normalized inputs satisfy $(\tilde{\epsilon}_i^r)^{(k)} \in [-1, 1]$ and $(\tilde{\Psi}^{el})^{(k)} \in [0, 1]$. Note that the rescaling process preserves the *zero strain equals zero energy* condition. Let us group the normalization factors in a diagonal matrix $G = \text{diag}(|\epsilon_1^r|_\infty, |\epsilon_2^r|_\infty, \dots, |\epsilon_8^r|_\infty)$. In the general 3D case, G will not have any zero diagonal entries if the training set is built. Here, $|\epsilon_1^r|_\infty, |\epsilon_{1,5-8}^r|_\infty$ will be zero, since we are dealing with the 2D case. One can simply ignore the normalization step for the zero vectors $\epsilon_{1,5-8}^r$. Thus, one can define

$$\tilde{\epsilon}^r = G^{-1} \epsilon^r. \quad (15)$$

Ultimately, we want to represent

$$\Psi^{el} = \hat{\Psi}^{el}(\boldsymbol{\varepsilon}^r) \approx \Psi^{surr}(\boldsymbol{\varepsilon}^r). \quad (16)$$

However, since both strains and energy are undergoing normalization process, we are actually training our surrogate models to predict

$$\tilde{\Psi}^{el} = \hat{\tilde{\Psi}}^{el}(\tilde{\boldsymbol{\varepsilon}}^r) \approx \tilde{\Psi}^{surr}(\tilde{\boldsymbol{\varepsilon}}^r). \quad (17)$$

Thus, for evaluation of the energy function, one simply has to perform

$$\Psi^{surr}(\boldsymbol{\varepsilon}^r) = |\boldsymbol{\Psi}^{el}|_{\infty} \tilde{\Psi}^{el}(\mathbf{G}^{-1}\boldsymbol{\varepsilon}^r). \quad (18)$$

Whilst, according to Equation (9), using chain rule, we get

$$\boldsymbol{\sigma}^r = \frac{\partial \Psi^{el}}{\partial \boldsymbol{\varepsilon}^r} = |\boldsymbol{\Psi}^{el}|_{\infty} \mathbf{G}^{-T} \frac{\partial \tilde{\Psi}^{el}}{\partial \tilde{\boldsymbol{\varepsilon}}^r}, \quad \mathbf{D} = \frac{\partial^2 \Psi^{el}}{\partial \boldsymbol{\varepsilon}^r \partial \boldsymbol{\varepsilon}^r} = |\boldsymbol{\Psi}^{el}|_{\infty} \mathbf{G}^{-T} \frac{\partial^2 \tilde{\Psi}^{el}}{\partial \tilde{\boldsymbol{\varepsilon}}^r \partial \tilde{\boldsymbol{\varepsilon}}^r} \mathbf{G}^{-1}. \quad (19)$$

It is clear now that once the surrogate for normalized energy is built, we can directly access both stress resultants and material tangent stiffness.

In the present work, the loss function is the sum of squared error. Thus, for a normalized rod strain input $\tilde{\boldsymbol{\varepsilon}}^r$, we get the pair of normalized energy data $(\tilde{\Psi}^{el})^{(i)}$ and predicted normalized energy $(\tilde{\Psi}_{NN}^{el})^{(i)} = \tilde{\Psi}^{el}(\tilde{\boldsymbol{\varepsilon}}^r)^{(i)}$. Then, the loss function is given by

$$E_{\Psi} = \sum_{i=1, n_{\text{input data}}} \left[(\tilde{\Psi}^{el})^{(i)} - (\tilde{\Psi}^{surr})^{(i)} \right]^2. \quad (20)$$

To adjust the weight parameters of the linear layers of the DNN, Adam optimizer [51] was employed, within a workflow that allows for either manual adjustment of the learning rates or automatic learning rate adjustment based on plateau detection. Training was carried so that each epoch is divided in batches, with user-defined size. To update the DNN's parameters, the required gradients of Equation (20) are automatically calculated by PyTorch.

4. Numerical Example: Results and Discussion

For this work, it is of interest an I-section member, as depicted in Figure 5. It is assumed usual elastic parameters for steel (Young's modulus $E = 200$ GPa and Poisson's coefficient $\nu = 0.25$), and a reference length of $L^r = 24$ cm. The choice of L^r is rather arbitrary, and was taken based on numerical auxiliary tests that showed that this length was sufficient to detect semi-waves from web/flange local buckling under simultaneous bending and compression.

For the high-fidelity model, simulations were made with the software FEAP v7.4 [52]. Standard displacement-based 8-noded (hexaedric) solid elements were used, in a regular mesh, as in Figure 5. Polyconvex isotropic Neo-Hookean material was employed (see FEAP's manual for details about the constitutive equation). The boundary conditions were applied using FEAP's "SPIN" command, which applies a rigid body motion to a selection of nodes. Simulations were carried incrementally, and terminated either when the load factor γ reached 1 or when the solver could not find further solutions by incrementing the load factor by a given minimum threshold. For this study, it was chosen as domain $\hat{u}_2, \hat{u}_3 \in [-24, +24]$ cm and $\hat{\theta}_1 \in [-90^\circ, +90^\circ]$.

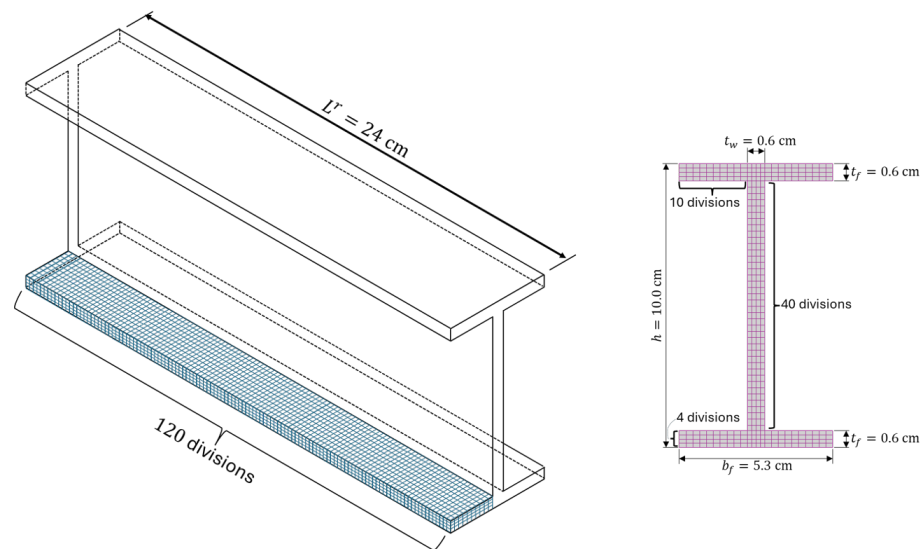


Figure 5. Mesh discretization for solid reference models.

Different architectures for the densely connected DNN were compared, see Figure 2. They were built as mentioned in Section 3. Primarily, the ELU activation function was used, for both surrogate models $\Psi_{(1)}^{surr}$ and $\Psi_{(2)}^{surr}$, using differently sized networks. In the optimally model, the alternatives with GELU and SiLU functions were evaluated. See Table 1 for details about the tested cases.

Table 1. Studied DNN architectures.

ID ¹	Layers	Width	Activation Function (ϕ)	Surrogate Model
L3_W32_ELU_psi	3	32	ELU	$\Psi_{(1)}^{surr} = \Psi^{NN}$
L5_W32_ELU_psi	5	32	ELU	$\Psi_{(1)}^{surr} = \Psi^{NN}$
L8_W32_ELU_psi	8	32	ELU	$\Psi_{(1)}^{surr} = \Psi^{NN}$
L3_W32_ELU_chi	3	32	ELU	$\Psi_{(2)}^{surr} = \chi^{NN} \Psi_{quad}$
L5_W32_ELU_chi	5	32	ELU	$\Psi_{(2)}^{surr} = \chi^{NN} \Psi_{quad}$
L8_W32_ELU_chi	8	32	ELU	$\Psi_{(2)}^{surr} = \chi^{NN} \Psi_{quad}$
L5_W32_GELU_chi	5	32	GELU	$\Psi_{(2)}^{surr} = \chi^{NN} \Psi_{quad}$
L5_W32_SiLU_chi	5	32	SiLU	$\Psi_{(2)}^{surr} = \chi^{NN} \Psi_{quad}$

¹ All layers are fully connected and of same width. Apart from the output layer, the activation functions act component-wise between subsequent layers.

In total, 98 procedurally generated and 82 randomly generated training inputs were used, alongside 60 randomly generated testing cases, as depicted in Tables A1 and A2 in Appendix A. Simulations were implemented so that the initial load factor increment was $\Delta\gamma = 2\%$ of the total prescribed displacements and rotations. In total, 7052 generalized strain- internal energy pairs were available for training, and 2173 for testing. A typical deformed configuration of those simulations can be seen in Figure 6.

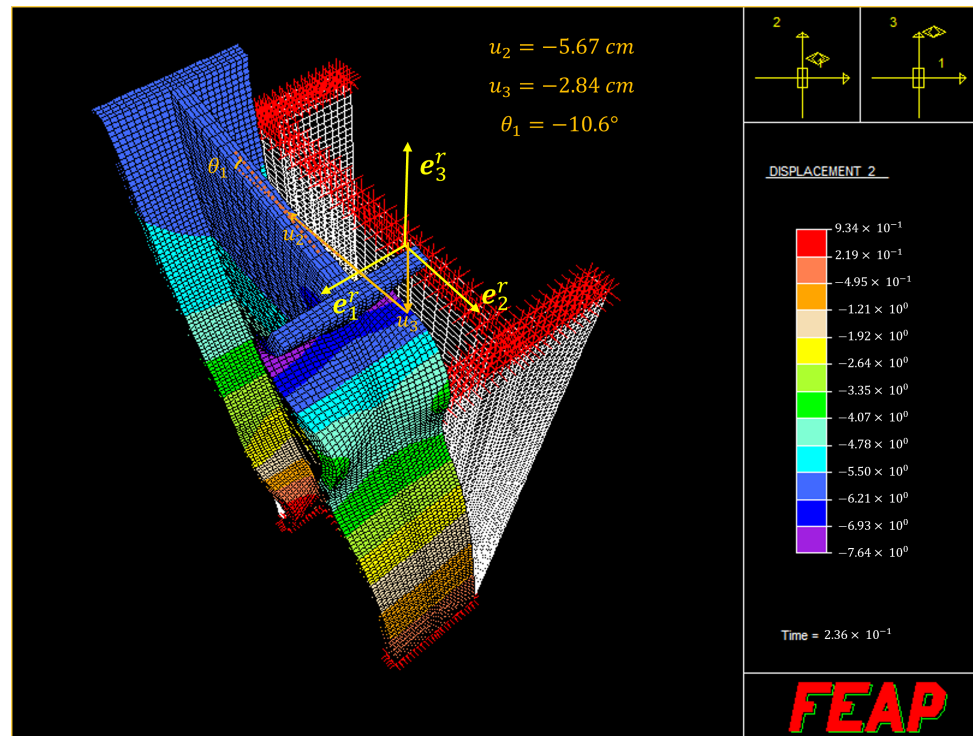


Figure 6. Last converged step for input 8 (see Table A1) (see Appendix A). Both deformed and undeformed meshes are shown. Direction 2 is aligned with the web. Note the presence of flange buckling.

For each training epoch, data pairs were divided into 100 pairs batches, randomly shuffled between epochs. The energy function and ϵ^r were normalized as described in Section 3.3. From the train dataset, we obtained $|\eta_2^r|_\infty = 1.1859$, $|\eta_3^r|_\infty = 0.5631$, $|\kappa_1^r|_\infty = 0.0321$ and $|\Psi^{el}|_\infty = 81,983$.

In Figure 7, training progress for each of the models from Table 1 is depicted. Training was carried until the measured errors in the test set began to increase for decreasing errors in the training set, which would indicate overfitting.

Figure 7, alongside with results from Figures 8 and 9, allow us to conclude that even the surrogate models with simpler architectures were able to represent the rod's energy remarkably well. Models with 5 and 8 layers behave similarly for the testing dataset. Also, in spite of smaller errors for training data in models using $\Psi_{(1)}^{surr}$, the error measure is virtually the same when comparing testing data to their equivalent $\Psi_{(2)}^{surr}$ counterparts.

Figures 8 and 9 depict the previsions obtained with our DNNs as compared with some of the given input (observation) data. As it can be seen, the surrogate models were able to represent the rod's energy remarkably well, including in large-strain configurations, even for simpler architectures (see how well models with 3 layers performed in the aforementioned figures). However, it is from models with 5 layers and above that accurate predictions are consistently made across the test set, see Figures 8 and 9. It is worth mentioning that the proposed solutions for imposing null strain-null energy condition worked as expected for all the cases.

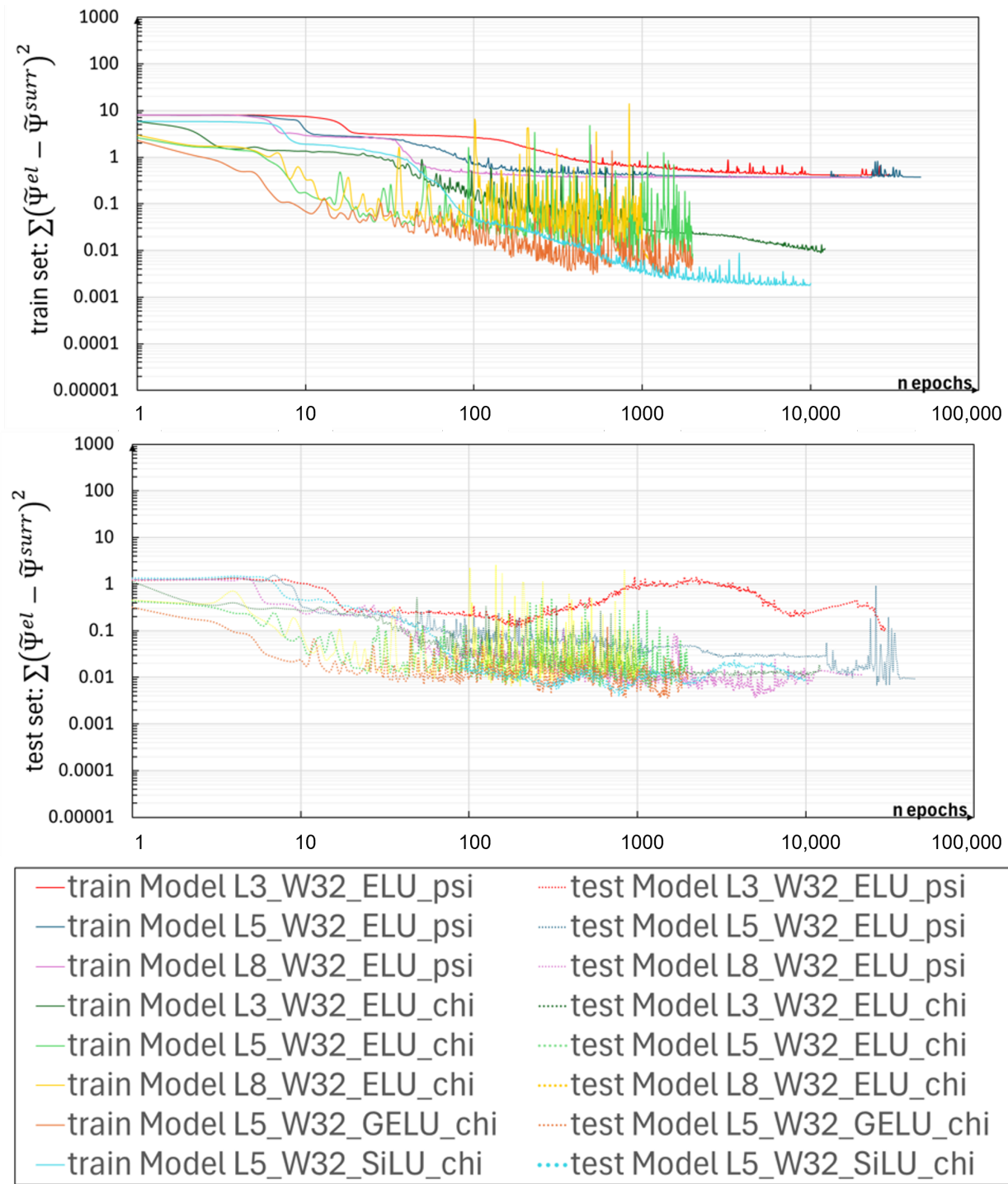


Figure 7. Training evolution of the different models. Training and testing sample sets error.

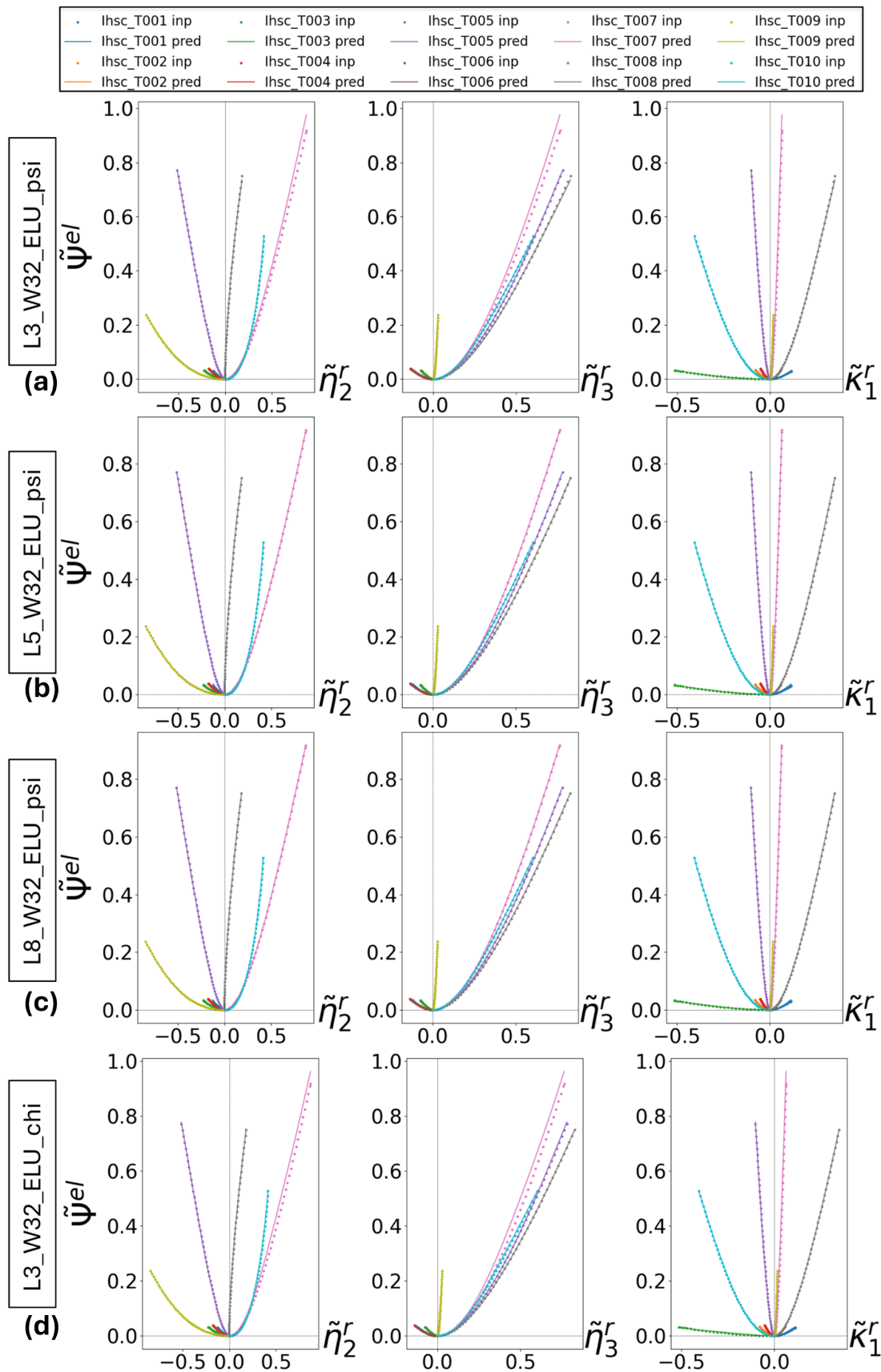


Figure 8. Prevision of test subset T001-T010, using model (a) L3_W32_ELU_psi, (b) L5_W32_ELU_psi, (c) L8_W32_ELU_psi, (d) L3_W32_ELU_chi. Dots are input data and lines are previsions.

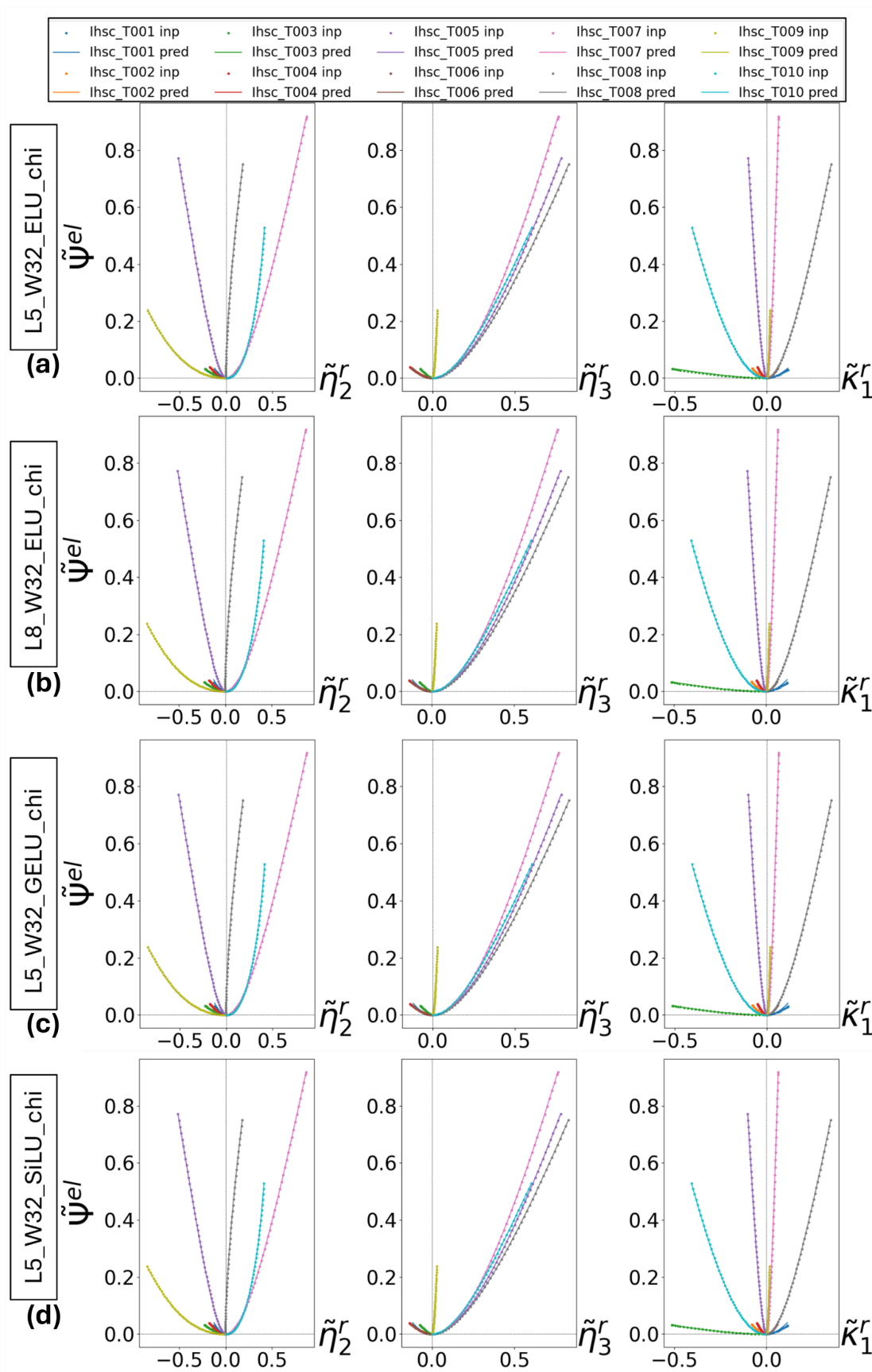


Figure 9. Prevision of test subset T001-T010, using model (a) L5_W32_ELU_chi, (b) L8_W32_ELU_chi, (c) L5_W32_GELU_chi, (d) L5_W32_SiLU_chi. Dots are input data and lines are previsions.

However, around zero strains, models using $\Psi_{(1)}^{surr}$ failed to comply with the non-negativity constraint, see Figure 10. For those, several regularization strategies (that are not reported here) were tested unsuccessfully. In light of the discussion from Section 3.1, the remedy was to use the alternative strategy $\Psi_{(2)}^{surr}$. Only then we were able to consistently generate models that predict well both large and small strain energy functions, while complying to the conditions mentioned above, see Figure 11. Therefore, from now on, the discussion will only consider surrogate models derived from $\Psi_{(2)}^{surr}$ approach.

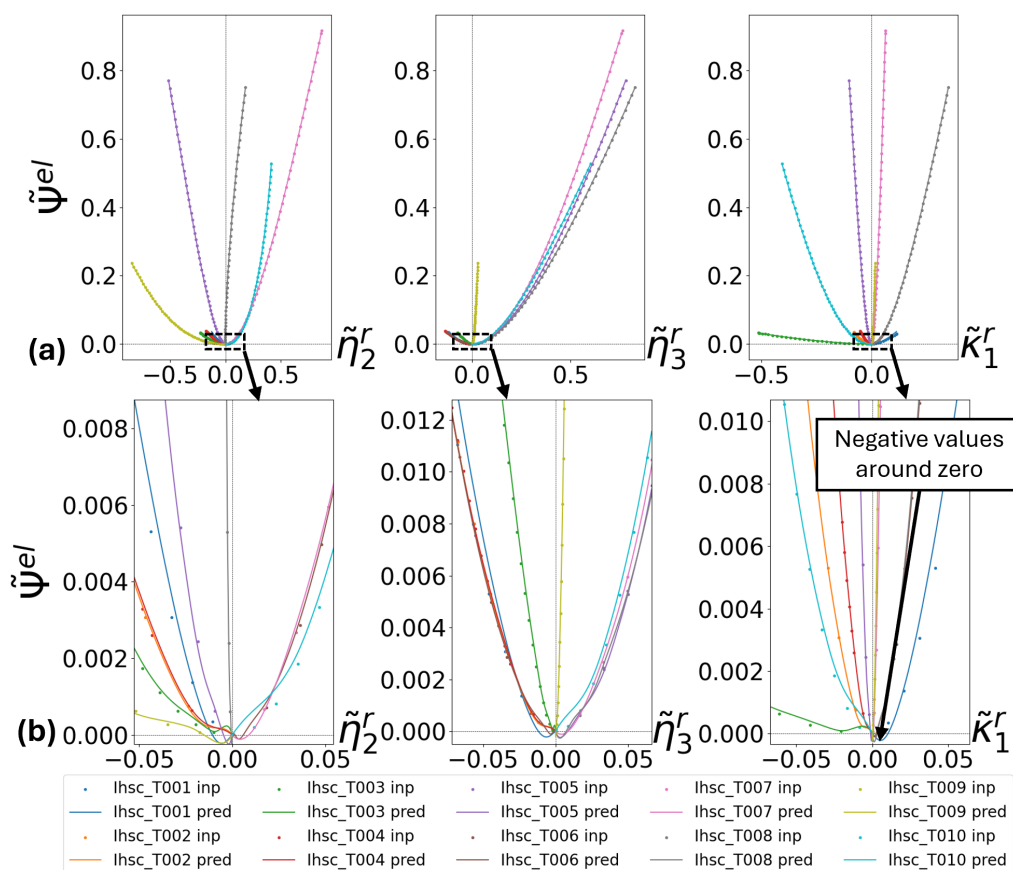


Figure 10. (a) Prevision of test subset T011-T020, using model L5_W32_ELU_psi. (b) Detail around zero strains.

Since model L5_W32_ELU_chi already had enough capacity to infer accurately the energy function, training of model L8_W32_ELU_chi was done with caution, since they could easily overfit. We highlight that little prediction capabilities was added when going from 5 to 8 layers. Given the sparsity of input data, we consider that the surrogate models we generated are remarkably accurate.

The optimal models are L5_W32_ELU_chi, L5_W32_GELU_chi and L5_W32_SiLU_chi, which have all the same amount of parameters and similar testing error. Among them, L5_W32_SiLU_chi performed slightly better, see Figure 7. Thus, in a future work with the fully functional rod model, the model L5_W32_SiLU_chi shall be employed. In Figures A1–A3, prevision curves for the complete training and testing sets are available.

Using Pytorch's automatic differentiation tools, in future works we intend to extract the calculation graph of the derivatives of Ψ^{el} for those particular models, and directly employ them in simulations with kinematically exact rods (recall Equation (9), those partial derivatives represent stress resultants and material tangent stiffness). While models with GELU and SiLu are inherently twice differentiable, studies about the second-order

differentiability of the ELU-based models will be done to ensure no numerical issues at runtime.

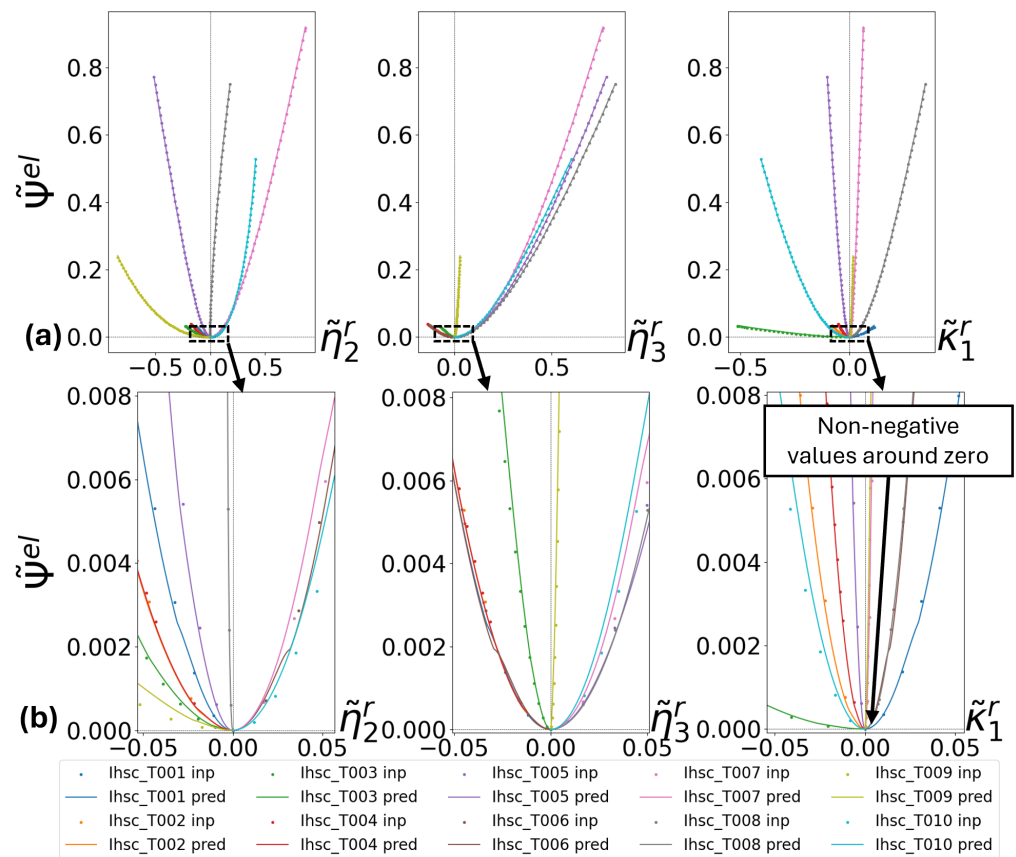


Figure 11. (a) Prevision of test subset T011-T020, using model L5_W32_ELU_chi. (b) Detail around zero strains.

5. Conclusions

In the current work

- a DNN-based surrogate model for rod constitutive equation was successfully built;
- a set of physical constraints for the surrogate model was enunciated in Section 3.1, alongside with measures that should be taken to guarantee compliance;
- conventional multi-layer perceptron architecture was compared to an innovative custom approach, in which the DNN acted as a multiplicative factor, applied to the usual quadratic energy function for rods. The latter proved to perform significantly better;
- Besides the DNN-based framework itself for rod constitutive equation, the effect of local bending and web/flange buckling was indirectly taken into account, as it is embedded in the member's elastic energy computed with refined (3D-solid) FE simulations, yet retaining the usual rod DOFs in the rod model. While the local buckling mode is not known, penalization of the energy function shall reflect the stiffness decrease during buckling process.
- The framework is ready to be employed to full 3D-frame rod simulations, wherein the computational gains over 3D-solid models will be much highlighted.

Author Contributions: Conceptualization, E.d.M.B.C. and A.I.; Formal analysis, M.P.K.; Software, M.P.K.; Supervision, E.d.M.B.C. and A.I.; Writing—original draft, M.P.K., E.d.M.B.C. and A.I.; Writing—review & editing, M.P.K., E.d.M.B.C. and A.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Fapesp (Sao Paulo State Research Foundation), Brazil, grants number 2023/16272-1, 2022/15644-0 and 2020/13362-1, CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Brazil, under the grant 313046/2021-2, as well as by ANR (Agence Nationale Recherche) in France under the grant ANR-20-CE46-0012-01 and IUF (Institut Universitaire France) under grant 1479.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

Acknowledgments: The authors acknowledge Fapesp, CNPq, ANR and IUF for the provided funding.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NN	Neural Network
DNN	Deep Neural Network
DOF	Degree-of-freedom
BC	Boundary condition
ML	Machine Learning

Appendix A. Solid Simulation BCs Inputs for Training and Testing of the DNN

In this appendix, the inputs for the example from Section 4 are provided. See Tables A1 and A2.

Table A1. Training inputs. Inputs 1–98 were procedurally generated, 99–180 were randomly generated.

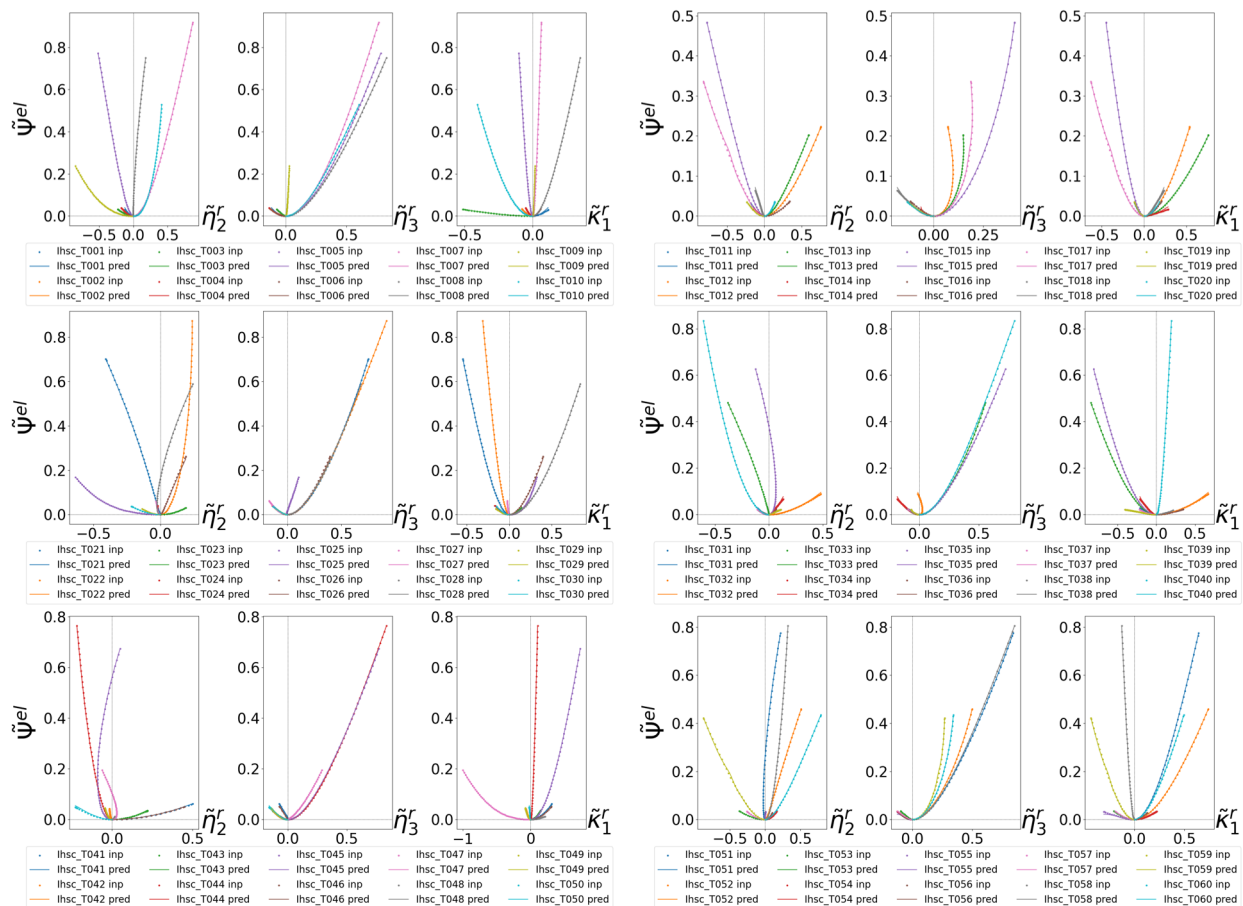
Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)	Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)	Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)	Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)	Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)
1	−24	0	0	37	−12	12	90	73	24	−24	90	109	24	19	48	145	19	−2	90
2	−24	−12	0	38	−12	−24	−90	74	24	24	−90	110	24	10	−83	146	24	−9	57
3	−24	12	0	39	−12	−24	90	75	24	24	90	111	21	10	−90	147	−24	−21	25
4	−24	−24	0	40	−12	24	−90	76	24	−12	−90	112	−24	19	−64	148	−1	−24	−75
5	−24	24	0	41	−12	24	90	77	24	−12	90	113	−14	−16	−90	149	24	−22	−34
6	−24	0	−45	42	0	−24	0	78	24	12	−90	114	−24	8	66	150	−17	−12	−90
7	−24	0	45	43	0	−24	−45	79	24	12	90	115	−20	24	89	151	−3	−24	15
8	−24	−12	−45	44	0	−24	45	80	24	0	−90	116	11	24	4	152	24	−4	−5
9	−24	−12	45	45	0	−24	−90	81	24	0	90	117	−3	−24	34	153	4	−24	−8
10	−24	12	−45	46	0	−12	45	82	24	−24	−45	118	24	10	81	154	3	−2	−90
11	−24	12	45	47	0	−12	−90	83	24	−24	45	119	24	−3	−32	155	−24	−2	6
12	−12	−24	0	48	0	−12	90	84	24	24	−45	120	11	−16	−90	156	24	3	−82
13	−12	24	0	49	0	0	−90	85	24	24	45	121	−24	12	−46	157	24	6	−19
14	−24	−24	−45	50	0	0	90	86	12	−24	0	122	−4	−13	−90	158	−24	9	−5
15	−24	−24	45	51	0	12	−90	87	12	24	0	123	−24	−24	42	159	−24	−3	15
16	−24	24	−45	52	0	12	90	88	24	−12	−45	124	−24	−3	−79	160	−24	0	18
17	−24	24	45	53	0	24	−90	89	24	−12	45	125	−8	−16	90	161	3	24	13
18	−24	0	−90	54	0	24	90	90	24	12	−45	126	24	−16	−52	162	−12	22	90
19	−24	0	90	55	0	24	−45	91	24	12	45	127	−10	15	90	163	18	24	−2
20	−24	−12	−90	56	0	24	45	92	24	0	−45	128	19	4	90	164	9	−14	−90
21	−24	−12	90	57	0	24	0	93	24	0	45	129	23	24	44	165	−24	−7	63
22	−24	12	−90	58	12	−24	−90	94	24	−24	0	130	24	−20	−75	166	3	−18	90
23	−24	12	90	59	12	−24	90	95	24	24	0	131	−13	22	90	167	12	−24	−86
24	−24	−24	−90	60	12	24	−90	96	24	−12	0	132	24	−18	−44	168	−24	−3	32
25	−24	−24	90	61	12	24	90	97	24	12	0	133	24	20	−7	169	7	−2	−90
26	−24	24	−90	62	12	−12	−90	98	24	0	0	134	−24	11	50	170	−24	−20	−69
27	−24	24	90	63	12	−12	90	99	−10	3	90	135	−7	10	−90	171	−15	−1	−90
28	−12	−24	−45	64	12	12	−90	100	−24	19	59	136	−14	12	90	172	−4	−11	90
29	−12	−24	45	65	12	12	90	101	24	−5	−11	137	−12	5	−90	173	3	−24	72
30	−12	24	−45	66	12	0	−90	102	−8	13	90	138	8	−24	−11	174	6	−14	−90
31	−12	24	45	67	12	0	90	103	−6	10	90	139	24	−22	78	175	−23	−24	54
32	−12	0	−90	68	12	−24	−45	104	−24	10	−79	140	−24	7	44	176	−9	−24	−80
33	−12	0	90	69	12	−24	45	105	6	−24	37	141	−16	−24	−45	177	24	−6	82
34	−12	−12	−90	70	12	24	−45	106	24	0	−51	142	−20	−15	−90	178	−24	−10	−84
35	−12	−12	90	71	12	24	45	107	−5	24	−83	143	11	6	90	179	−15	−21	90
36	−12	12	−90	72	24	−24	−90	108	24	−19	42	144	6	1	90	180	−17	24	39

Table A2. Testing inputs, randomly generated.

Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)	Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)	Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)	Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)	Id	$\hat{u}_2$ (cm)	$\hat{u}_3$ (cm)	$\hat{\theta}_1$ (°)
T01	−24	−17	46	T13	−3	12	90	T25	−24	−2	29	T37	3	−12	−90	T49	0	−24	−41
T02	−22	−24	−49	T14	−24	−9	67	T26	−12	24	75	T38	−24	−12	76	T50	−24	−18	−7
T03	5	−3	−90	T15	−11	24	−57	T27	−3	−24	−10	T39	24	−10	−89	T51	−20	24	64
T04	−24	−22	−27	T16	24	−9	72	T28	−24	17	85	T40	−24	21	18	T52	−12	22	90
T05	−11	24	−9	T17	−10	20	−90	T29	−8	−18	−90	T41	24	2	82	T53	−24	−6	−66
T06	18	−24	35	T18	−24	−22	75	T30	−24	−22	−90	T42	4	−24	−38	T54	4	−14	90
T07	22	24	6	T19	−24	−17	−52	T31	3	−15	−90	T43	24	−20	83	T55	17	−14	−90
T08	−8	24	31	T20	5	−19	90	T32	0	5	90	T44	−10	24	9	T56	6	−24	89
T09	−24	1	1	T21	9	24	−52	T33	19	17	−90	T45	−24	16	66	T57	−21	−15	−90
T10	24	12	−37	T22	19	24	−28	T34	24	−23	−68	T46	23	1	90	T58	14	24	−11
T11	−24	−24	−73	T23	20	−21	90	T35	24	15	−73	T47	22	0	−90	T59	−21	24	−66
T12	16	16	90	T24	12	−24	−86	T36	−11	−9	90	T48	−15	−10	90	T60	12	24	68

Appendix B. DNN Results for Training and Testing Sets Using Model L5_W32_SiLU_chi

Figures A1–A3 depict the prevision values for the generated testing and training datasets, using model L5_W32_SiLU_chi (see Tables 1, A1 and A2 for details).

**Figure A1.** Prevision of test dataset, using model L5_W32_SiLU_chi.

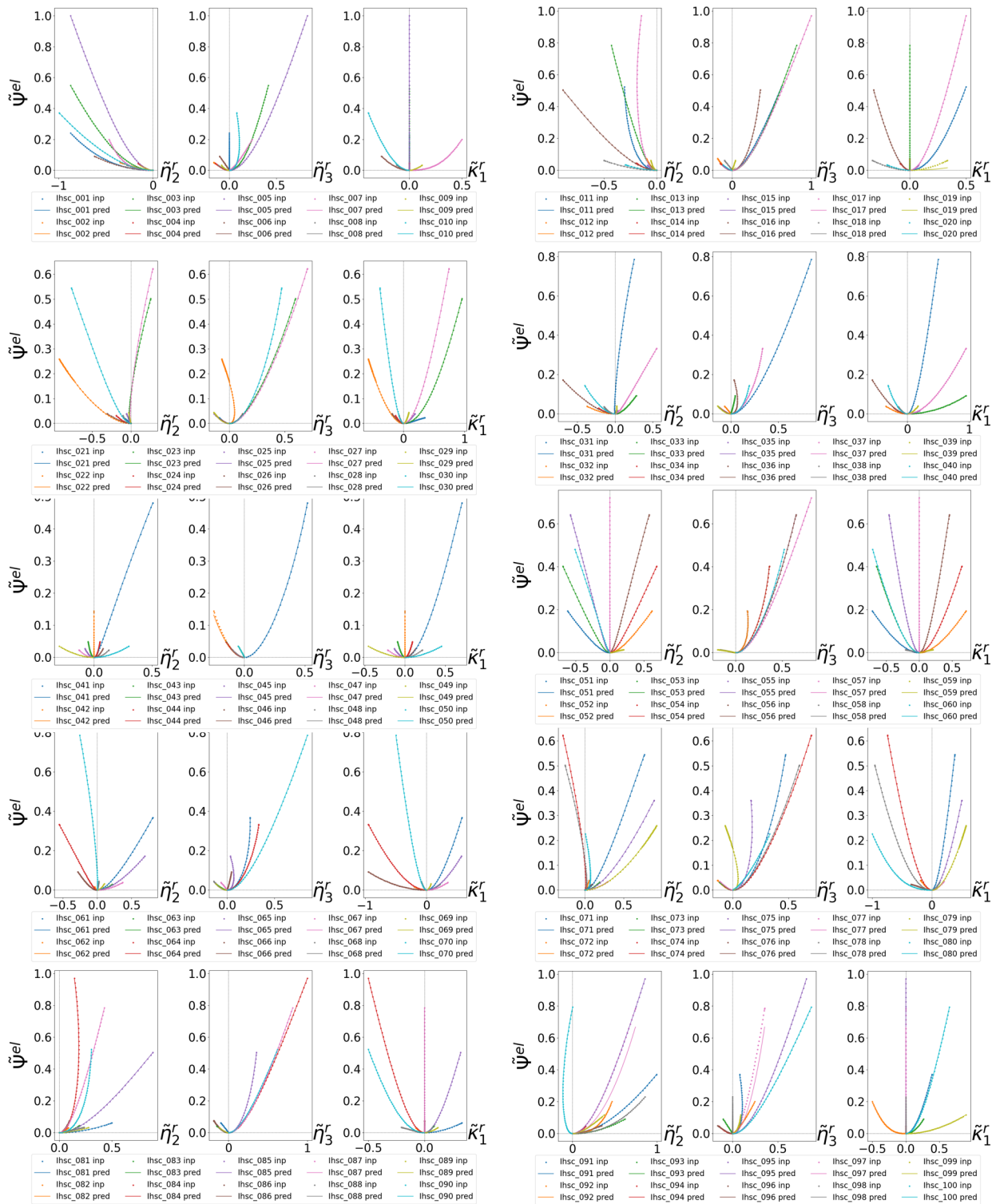


Figure A2. Prevision of train dataset, using model L5_W32_SiLU_chi, inputs 1 to 100.

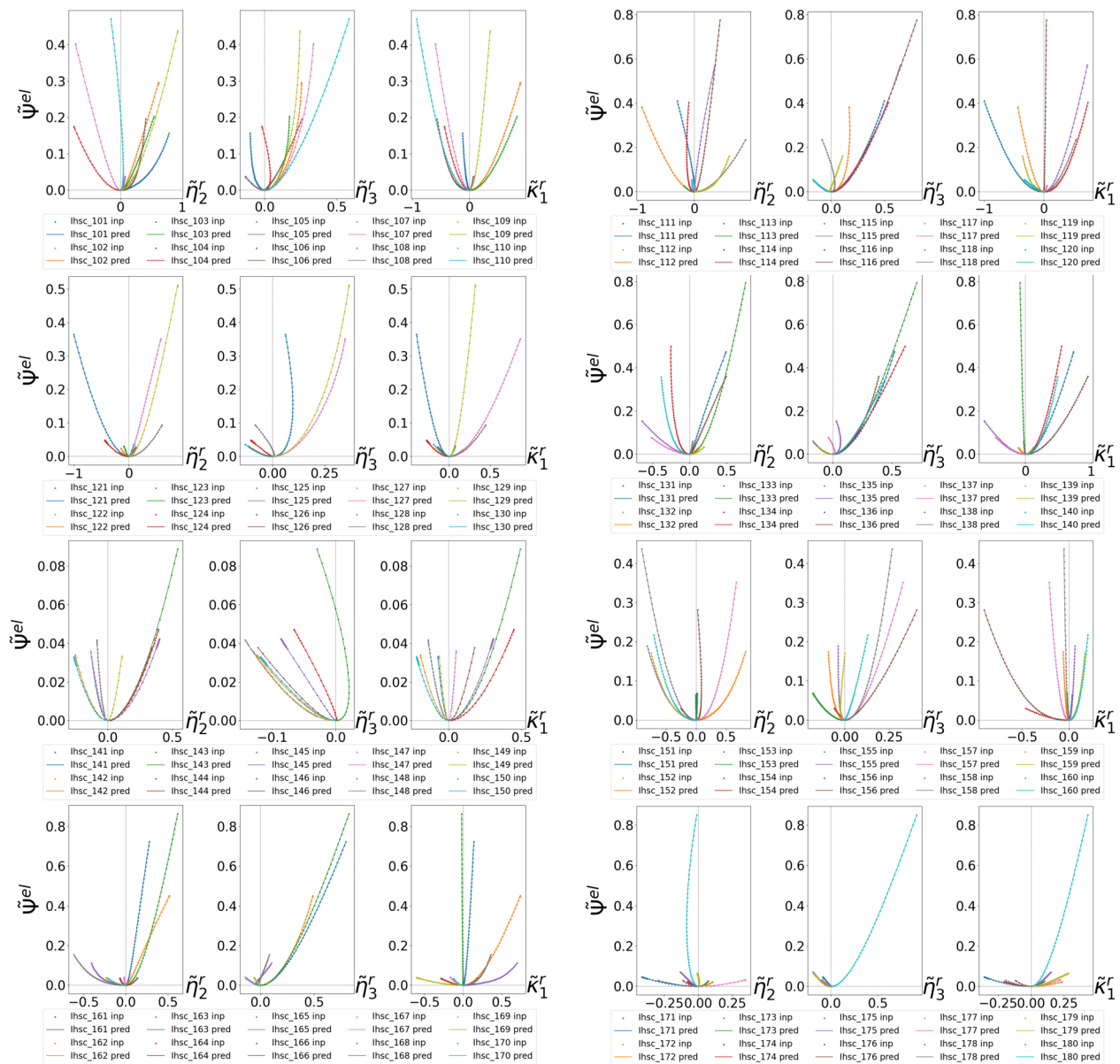


Figure A3. Prevision of train dataset, using model L5_W32_SiLU_chi, inputs 101 to 181.

References

1. Reissner, E. On one-dimensional finite-strain beam theory: The plane problem. *Z. Angew. Math. Phys. Zamp* **1972**, *23*, 795–804. [\[CrossRef\]](#)
2. Argyris, J. An excursion into large rotations. *Comput. Methods Appl. Mech. Eng.* **1982**, *32*, 85–155. [\[CrossRef\]](#)
3. Ibrahimbegovic, A. On the choice of finite rotation parameters. *Comput. Methods Appl. Mech. Eng.* **1997**, *149*, 49–71. [\[CrossRef\]](#)
4. Simo, J.C. A finite strain beam formulation. The three-dimensional dynamic problem. Part I. *Comput. Methods Appl. Mech. Eng.* **1985**, *49*, 55–70. [\[CrossRef\]](#)
5. Simo, J.; Vu-Quoc, L. A three-dimensional finite-strain rod model. part II: Computational aspects. *Comput. Methods Appl. Mech. Eng.* **1986**, *58*, 79–116. [\[CrossRef\]](#)
6. Pimenta, P.M.; Yojo, T. Geometrically Exact Analysis of Spatial Frames. *Appl. Mech. Rev.* **1993**, *46*, S118–S128. [\[CrossRef\]](#)
7. Ibrahimbegović, A.; Frey, F. Finite element analysis of linear and non-linear planar deformations of elastic initially curved beams. *Int. J. Numer. Methods Eng.* **1993**, *36*, 3239–3258. [\[CrossRef\]](#)
8. Vlasov, V.Z. *Thin-Walled Elastic Beams*, 2nd ed.; Israel Program for Scientific Translation: Jerusalem, Israel, 1961.
9. Simo, J.C.; Vu-Quoc, L. A Geometrically-exact rod model incorporating shear and torsion-warping deformation. *Int. J. Solids Struct.* **1991**, *27*, 371–393. [\[CrossRef\]](#)

10. Pimenta, P.M.; Campello, E.M.B. Geometrically nonlinear analysis of thin-walled space frames. In Proceedings of the II ECCM (European Conference on Computational Mechanics), Cracow, Poland, 26–29 June 2001; p. 20.
11. Campello, E.M.B. *Análise Não-Linear de Perfis Metálicos Conformados a Frio*. Master's Thesis, Escola Politécnica, São Paulo, Brazil, 2000. [\[CrossRef\]](#)
12. Gruttmann, F.; Sauer, R.; Wagner, W. Theory and numerics of three-dimensional beams with elastoplastic material behaviour. *Int. J. Numer. Methods Eng.* **2000**, *48*, 33. [\[CrossRef\]](#)
13. Gonçalves, R. An assessment of the lateral-torsional buckling and post-buckling behaviour of steel I-section beams using a geometrically exact beam finite element. *Thin-Walled Struct.* **2019**, *143*, 106222. [\[CrossRef\]](#)
14. Campello, E.M.B.; Lago, L.B. Effect of higher order constitutive terms on the elastic buckling of thin-walled rods. *Thin-Walled Struct.* **2014**, *77*, 8–16. [\[CrossRef\]](#)
15. Kassab, M.P.; Campello, E.M.B.; Pimenta, P.M. Advances on kinematically exact rod models for thin-walled open-section members: Consistent warping function and nonlinear constitutive equation. *Comput. Methods Appl. Mech. Eng.* **2023**, *407*, 115933. [\[CrossRef\]](#)
16. Kassab, M.P.; Campello, E.M.B.; Ibrahimbegovic, A. A kinematically-exact reduced-order rod model for elastoplastic failure in thin-walled members. In Proceedings of the 16th World Congress on Computational Mechanics and 4th Pan American Congress on Computational Mechanics, Vancouver, BC, USA, 21–26 July 2024. [\[CrossRef\]](#)
17. Kassab, M.P.; Campello, E.M.B.; Ibrahimbegovic, A. A geometrically exact thin-walled rod model with warping and stress-resultant-based plasticity obtained with a two-level computational approach. *Comput. Methods Appl. Mech. Eng.* **2025**, *433*, 117497. [\[CrossRef\]](#)
18. Pimenta, P.M.; Campello, E.M.B. A fully nonlinear multi-parameter rod model incorporating general cross-sectional in-plane changes and out-of-plane warping. *Lat. Am. J. Solids Struct.* **2003**, *1*, 119–140.
19. Dasambiagio, E.R.; Pimenta, P.M.; Campello, E.M.B. A finite strain rod model that incorporates general cross section deformation and its implementation by the Finite Element Method; Brazilian Society of Mechanical Sciences and Engineering. *Solid Mech.* **2009**, *1*, 145–168.
20. Gonçalves, R.; Ritto-Corrêa, M.; Camotim, D. A large displacement and finite rotation thin-walled beam formulation including cross-section deformation. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 1627–1643. [\[CrossRef\]](#)
21. Thakolkaran, P.; Joshi, A.; Zheng, Y.; Flaschel, M.; De Lorenzis, L.; Kumar, S. NN-EUCLID: Deep-learning hyperelasticity without stress data. *J. Mech. Phys. Solids* **2022**, *169*, 105076. [\[CrossRef\]](#)
22. Flaschel, M.; Kumar, S.; De Lorenzis, L. Automated discovery of generalized standard material models with EUCLID. *Comput. Methods Appl. Mech. Eng.* **2023**, *405*, 115867. [\[CrossRef\]](#)
23. Kissas, G.; Mishra, S.; Chatzi, E.; De Lorenzis, L. The language of hyperelastic materials. *Comput. Methods Appl. Mech. Eng.* **2024**, *428*, 117053. [\[CrossRef\]](#)
24. Cao, D.; Bai, G.C. DNN-Based Surrogate Modeling-Based Feasible Performance Reliability Design Methodology for Aircraft Engine. *IEEE Access* **2020**, *8*, 229201–229218. [\[CrossRef\]](#)
25. Zhang, X.; Xie, F.; Ji, T.; Zhu, Z.; Zheng, Y. Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. *Comput. Methods Appl. Mech. Eng.* **2021**, *373*, 113485. [\[CrossRef\]](#)
26. Kontou, M.G.; Asouti, V.G.; Giannakoglou, K.C. DNN surrogates for turbulence closure in CFD-based shape optimization. *Appl. Soft Comput.* **2023**, *134*, 110013. [\[CrossRef\]](#)
27. Jaszczur, M.; Borowski, M.; Halibart, J.; Zwolińska-Gładys, K.; Marczak, P. Optimization of the Small Wind Turbine Design—Performance Analysis. *Computation* **2024**, *12*, 215. [\[CrossRef\]](#)
28. Shin, S.; Ko, B.; So, H. Noncontact thermal mapping method based on local temperature data using deep neural network regression. *Int. J. Heat Mass Transf.* **2022**, *183*, 122236. [\[CrossRef\]](#)
29. Quintana-Ruiz, O.; Campello, E.M.B. A methodology to predict the effective thermal conductivity of a granular assembly using deep learning. In Proceedings of the XLIII Ibero-Latin American Congress on Computational Methods in Engineering, Foz do Iguaçu, Brazil, 21–25 November 2022; p. 11.
30. Naffer-Chevassier, K.; Vuyst, F.D.; Goardou, Y. Enhanced Drag Force Estimation in Automotive Design: A Surrogate Model Leveraging Limited Full-Order Model Drag Data and Comprehensive Physical Field Integration. *Computation* **2024**, *12*, 207. [\[CrossRef\]](#)
31. Nguyen, H.V.; Bui-Thanh, T. TNet: A Model-Constrained Tikhonov Network Approach for Inverse Problems. *arXiv* **2022**, arXiv:2105.12033. [\[CrossRef\]](#)
32. Van Truong, V.; Noeiaghdam, S.; Sidorov, D.; Dreglea, A.; Wang, L. Solving Nonlinear Energy Supply and Demand System Using Physics-Informed Neural Networks. *Computation* **2025**, *13*, 13. [\[CrossRef\]](#)
33. Saha, S.; Gan, Z.; Cheng, L.; Gao, J.; Kafka, O.L.; Xie, X.; Li, H.; Tajdari, M.; Kim, H.A.; Liu, W.K. Hierarchical Deep Learning Neural Network (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering. *Comput. Methods Appl. Mech. Eng.* **2021**, *373*, 113452. [\[CrossRef\]](#)

34. Borzunov, S.V.; Semenov, M.E.; Zybin, E.Y.; Zheltov, S.Y.; Kosyanchuk, V.V.; Barsukov, A.I. A Simple Model of Turbine Control Under Stochastic Fluctuations of Internal Parameters. *Computation* **2025**, *13*, 27. [\[CrossRef\]](#)
35. Su, A.; Cheng, J.; Li, X.; Zhong, Y.; Li, S.; Zhao, O.; Jiang, K. Unified machine-learning-based design method for normal and high strength steel I-section beam–columns. *Thin-Walled Struct.* **2024**, *199*, 111835. [\[CrossRef\]](#)
36. Liu, J.; Li, S.; Guo, J.; Xue, S.; Chen, S.; Wang, L.; Zhou, Y.; Luo, T.X. Machine learning (ML) based models for predicting the ultimate bending moment resistance of high strength steel welded I-section beam under bending. *Thin-Walled Struct.* **2023**, *191*, 111051. [\[CrossRef\]](#)
37. Ly, H.B.; Le, T.T.; Le, L.M.; Tran, V.Q.; Le, V.M.; Vu, H.L.T.; Nguyen, Q.H.; Pham, B.T. Development of Hybrid Machine Learning Models for Predicting the Critical Buckling Load of I-Shaped Cellular Beams. *Appl. Sci.* **2019**, *9*, 5458. [\[CrossRef\]](#)
38. Degtyarev, V.V.; Tsavdaridis, K.D. Buckling and ultimate load prediction models for perforated steel beams using machine learning algorithms. *J. Build. Eng.* **2022**, *51*, 104316. [\[CrossRef\]](#)
39. Rabi, M.; Jweihan, Y.S.; Abarkan, I.; Ferreira, F.P.V.; Shamass, R.; Limbachiya, V.; Tsavdaridis, K.D.; Santos, L.F.P. Machine learning-driven web-post buckling resistance prediction for high-strength steel beams with elliptically-based web openings. *Results Eng.* **2024**, *21*, 101749. [\[CrossRef\]](#)
40. Dai, Y.; Roy, K.; Fang, Z.; Chen, B.; Raftery, G.M.; Lim, J.B. A novel machine learning model to predict the moment capacity of cold-formed steel channel beams with edge-stiffened and un-stiffened web holes. *J. Build. Eng.* **2022**, *53*, 104592. [\[CrossRef\]](#)
41. Hajdú, G.; Bektaş, N.; Müller, A. Machine learning models for the elastic-critical buckling moment of sinusoidal corrugated web beam. *Results Eng.* **2024**, *23*, 102371. [\[CrossRef\]](#)
42. Karabini, M.; Karampinis, I.; Rousakis, T.; Iliadis, L.; Karabinis, A. Machine Learning Ensemble Methodologies for the Prediction of the Failure Mode of Reinforced Concrete Beam–Column Joints. *Information* **2024**, *15*, 647. [\[CrossRef\]](#)
43. Nariman, N.; Hamdia, K.; Ramadan, A.; Sadaghian, H. Optimum Design of Flexural Strength and Stiffness for Reinforced Concrete Beams Using Machine Learning. *Appl. Sci.* **2021**, *11*, 8762. [\[CrossRef\]](#)
44. Shatnawi, A.; Alkassar, H.M.; Al-Abdaly, N.M.; Al-Hamdany, E.A.; Bernardo, L.F.A.; Imran, H. Shear Strength Prediction of Slender Steel Fiber Reinforced Concrete Beams Using a Gradient Boosting Regression Tree Method. *Buildings* **2022**, *12*, 550. [\[CrossRef\]](#)
45. Truong, V.H.; Pham, H.A.; Van, T.H.; Tangaramvong, S. Evaluation of machine learning models for load-carrying capacity assessment of semi-rigid steel structures. *Eng. Struct.* **2022**, *273*, 115001. [\[CrossRef\]](#)
46. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2019.
47. Ibrahimbegovic, A. *Nonlinear Solid Mechanics*; Springer: Dordrecht, The Netherlands, 2009; Volume 160. [\[CrossRef\]](#)
48. Ibrahimbegovic, A.; Mejia-Nava, R.A. *Structural Engineering: Models and Methods for Statics, Instability and Inelasticity*; Springer International Publishing: Berlin/Heidelberg, Germany, 2023; Volume 100. [\[CrossRef\]](#)
49. Ibrahimbegovic, A.; Mejia-Nava, R.; Ljukovac, S. Reduced model for fracture of geometrically exact planar beam: Non-local variational formulation, ED-FEM approximation and operator split solution. *Int. J. Numer. Methods Eng.* **2024**, *125*, e7369. [\[CrossRef\]](#)
50. Jaka Dujc.; Brank, B.; Ibrahimbegovic, A. Multi-scale computational model for failure analysis of metal frames that includes softening and local buckling. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 1371–1385. [\[CrossRef\]](#)
51. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
52. Taylor, R.L. FEAP—Finite Element Analysis Program: Version 8.4 User Manual, 2014. Available online: <http://projects.ce.berkeley.edu/feap/> (accessed on 12 March 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.