Direct Hybrid Collocation PINN (DHC-PINN / IE-PINN) for Inverse Identification of Spatially-Varying Elastic Moduli in Additively Manufactured Lattice Structures

A Complete Mathematical Lecture: From Continuum Mechanics to Saddle-Point Optimisation

Abstract

This lecture presents, from first principles, a Physics-Informed Neural Network (PINN) framework under the *Direct Hybrid Collocation* (DHC) paradigm for the inverse identification of spatially-varying Young's moduli $E(x, y)$ in three additively manufactured specimens: an 80% Gyroid, a 60% Gyroid, and an 80% Lines infill structure. The core architectural innovation over standard PINN formulations is the decoupling of the data stream from the physics stream: sparse Digital Image Correlation (DIC) blocks serve as *Dirichlet anchors*, while a dense Sobol quasi-random collocation cloud enforces the 2-D Cauchy Momentum PDE across the full specimen domain *without any associated displacement labels*. This prevents the non-physical extrapolation (strains exceeding 26,000%) observed when using radial basis function (RBF) pre-interpolation. Training is posed as a constrained saddle-point problem solved by simultaneous Adam primal descent and Adam dual ascent over six adaptively learned Lagrange multipliers, augmented by a Tikhonov smoothness regulariser on the modulus field and a learning-rate scheduler. Every mathematical detail — kinematics, constitutive law, dimensional restoration, Fourier embedding, collocation design, and optimisation theory — is derived in full.

Department of Mechanical Engineering · Computational Mechanics & Physics-Informed Machine Learning

Based on the `A_DHC.py` implementation. References: Raissi et al. (2019), Haghighat et al. (2021).

Contents

1 Motivation: Why Direct Hybrid Collocation?

1.1 The Sparse-Data Problem in DIC-Based Inverse Elasticity

Digital Image Correlation cameras measure the displacement field at a finite number of *tracking blocks* — in the present experiments, 8 DIC blocks are distributed across a specimen whose full domain is $\Omega = [0, W] \times [0, H]$. To enforce the momentum balance PDE, one needs displacement gradients $\partial u/\partial x$, $\partial u/\partial y$, etc., *everywhere in* Ω, not just at the 8 anchor positions. The challenge is to interpolate physically between anchors.

1.2 Failure Mode of RBF Pre-Interpolation

A naive approach uses Radial Basis Functions (RBFs) to fit a smooth surface through the 8 anchor values and then differentiates that surface for strains. Because RBFs treat each block as an isolated peak with no knowledge of mechanics, they are free to extrapolate with unconstrained gradients. In practice this produces:

- Predicted displacements $\sim 9{,}000\,\text{mm}$ (specimen is $\sim 100\,\text{mm}$ long).
- Predicted strains $\sim 26{,}000\%$ — physically impossible for an elastic specimen.

1.3 The DHC Paradigm

The Direct Hybrid Collocation method replaces the RBF pre-interpolation with the neural network itself. The key idea is:

Core DHC Philosophy

Stream A — Data (anchors): The 8 DIC blocks act as sparse Dirichlet constraints. The network is penalised if its displacement predictions deviate from the measured values at these points.

Stream B — Physics (collocation): $N_c = 10{,}000$ Sobol quasi-random points are scattered across the *entire* domain Ω. At these points the network receives *no displacement label*. It is penalised only if its predictions violate the 2-D Cauchy Momentum PDE.

Result: Any displacement field that fits the anchors but produces a non-physical strain gradient (e.g. 26,000%) generates a huge stress divergence at thousands of collocation points, which the optimiser immediately corrects. The PDE acts as a *global smoothness constraint* rooted in continuum mechanics.

2 Problem Statement and Material Parameters

2.1 Specimen Domain

Three specimens are studied, each with a rectangular domain

$$\Omega = [0, W] \times [0, H], \qquad H = 100\,\text{mm}.$$

The geometry and reference material constants for each specimen are:

Material Parameters per Specimen

Specimen	W (mm)	t (mm)	$A = W \times t$ (mm^2)	ν	ρ (kg/m^3)	E_{ref} (GPa)
80% Gyroid	30.30	4.94	149.68	0.35	1000	1.8
60% Gyroid	30.04	4.98	149.60	0.36	860	1.3
80% Lines	29.745	5.00	148.73	0.34	1050	2.25

Remark 2.1. *Setting $\rho \to 0$ in the code reduces the PDE to static equilibrium $\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}$. A non-zero density activates the inertia term $\rho\,\ddot{\mathbf{u}}$, turning the PDE into an elastodynamic wave equation. Both cases are supported by a single conditional in* `compute_pde_residual`.

2.2 Governing Equations of 2-D Elasticity

2.2.1 Linear Momentum Balance

In the absence of body forces other than inertia, Newton's second law per unit volume yields the strong-form PDE on $\Omega \times [t_s, t_e]$:

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}, t) = \rho\,\ddot{\mathbf{u}}(\mathbf{x}, t), \tag{1}$$

where $\mathbf{u} = (u, v)^\top$ is the displacement field and $\boldsymbol{\sigma}$ is the symmetric Cauchy stress tensor. Component-wise:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} = \rho\,\ddot{u}, \tag{2}$$

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} = \rho\,\ddot{v}. \tag{3}$$

When $\rho = 0$ these reduce to the static equilibrium equations $\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}$.

2.2.2 Kinematics: Infinitesimal Strain Tensor

Under small deformations:

$$\boldsymbol{\varepsilon} = \tfrac{1}{2}\big(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top\big), \tag{4}$$

giving the engineering strain components

$$\varepsilon_{xx} = \frac{\partial u}{\partial x}, \qquad \varepsilon_{yy} = \frac{\partial v}{\partial y}, \qquad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{5}$$

2.2.3 Constitutive Law: Plane-Stress Hooke's Law with Spatially-Varying Modulus

For an isotropic linear-elastic solid under plane stress the constitutive relation is:

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{pmatrix} = \underbrace{\frac{E(x,y)}{1-\nu^2}}_{\text{factor}} \begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{pmatrix} \begin{pmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{pmatrix}. \tag{6}$$

The spatially-varying Young's modulus $E(x, y) > 0$ is the primary *unknown* to be identified by the PINN. The shear modulus follows analytically:

$$G(x, y) = \frac{E(x, y)}{2(1 + \nu)}. \tag{7}$$

Expanding eq. (6):

$$\sigma_{xx} = \frac{E}{1-\nu^2}(\varepsilon_{xx} + \nu\varepsilon_{yy}), \tag{8}$$

$$\sigma_{yy} = \frac{E}{1-\nu^2}(\varepsilon_{yy} + \nu\varepsilon_{xx}), \tag{9}$$

$$\tau_{xy} = G\,\gamma_{xy}. \tag{10}$$

3 Data Acquisition, Calibration, and Pre-Processing

3.1 Dual-Sensor Architecture

Two independent measurement streams are acquired and subsequently synchronised:

(a) **DIC (Digital Image Correlation):** Records full-field displacements (u_i, v_i) and engineering strains $(\varepsilon_{xx,i}, \varepsilon_{yy,i}, \gamma_{xy,i})$ at each of the K tracking blocks $\{\mathbf{x}_i\}$ at frame rate $f_{\text{DIC}} = 10\,\text{Hz}$. Columns required: `x_phys`, `y_phys`, `u_train`, `v_train`, `exx_AVG`, `eyy_AVG`, `exy_AVG`.

(b) **UTM (Universal Testing Machine):** Records the scalar applied load $P(t)$ at a higher sampling rate. Columns required: `Time`, `Load`.

3.2 Temporal Stride Sub-Sampling

To reduce the dataset size without discarding physical content, only every $s_f = 15^{\text{th}}$ DIC frame is retained:

$$\mathcal{I}_{\text{stride}} = \{k \in \mathbb{Z}_{\geq 0} : k \bmod s_f = 0\}. \tag{11}$$

3.3 Hookean Regime Isolation via Rolling R^2 Analysis

The constitutive model eq. (6) is valid only in the *linear elastic* regime. A rolling R^2 criterion identifies the maximal Hookean sub-interval.

Definition 3.1 (Hookean Regime)**.** *Let $\{t_i, P_i\}_{i=0}^{N_{\text{UTM}}-1}$ be the filtered load–time series up to peak load. For a sliding window of width $w = 5$ samples starting at index i, define the coefficient of determination as*

$$R_i^2 = \left[\frac{\sum_{j=i}^{i+w-1} (t_j - \bar{t}_i)(P_j - \bar{P}_i)}{\sqrt{\sum_{j=i}^{i+w-1} (t_j - \bar{t}_i)^2 \sum_{j=i}^{i+w-1} (P_j - \bar{P}_i)^2}} \right]^2, \tag{12}$$

*where $\bar{t}_i, \bar{P}_i$ are the window means. The **Hookean regime** is the largest contiguous block of indices satisfying $R_i^2 \geq R^2_{\min} = 0.999$.*

A safety margin of w indices is trimmed from both ends of the detected block to guarantee strict interiority, yielding the temporal bounds $[t_s, t_e]$. All DIC frames outside this window are discarded:

$$\mathcal{D}_{\text{Hookean}} = \{(\mathbf{x}_i, t_k) : t_s \leq t_k \leq t_e\}. \tag{13}$$

3.4 Savitzky–Golay Displacement Smoothing

Per-frame displacement noise is suppressed via a Savitzky–Golay polynomial filter applied independently to each tracking block. For window length $w_{\text{sg}} = 11$ and polynomial order $p = 3$, the estimate at sample k within a block is:

$$\hat{u}_k = \sum_{j=-m}^{m} c_j \, u_{k+j}, \qquad m = \tfrac{w_{\text{sg}}-1}{2}, \tag{14}$$

where the convolution coefficients $\{c_j\}$ are the least-squares projection onto the polynomial basis $\{1, j, j^2, j^3\}$ over the w_{sg}-point stencil. This operation preserves polynomial signal content up to degree p while attenuating higher-frequency noise.

3.5 Coordinate Calibration

DIC outputs pixel coordinates; the network operates in physical units (mm). An automatic scale factor is computed as:

$$s_x = \frac{W_{\text{phys}}}{\max_i(x_i^{\text{csv}})}, \qquad s_y = \frac{H_{\text{phys}}}{\max_i(y_i^{\text{csv}})}, \tag{15}$$

so that $x_{\text{mm}} = s_x \cdot x_{\text{csv}}$ maps the data range to the specimen dimensions. These can be overridden by user-specified constants `COORD_SCALE_X` and `COORD_SCALE_Y`.

3.6 Displacement Zeroing: Rigid-Body Subtraction

The bottom grip is clamped and therefore undergoes zero net displacement. To enforce this, the mean displacement at the *bottom 15%* of the specimen is computed and subtracted:

$$u_{\text{mm},i} = \big(u_{\text{raw},i} - \bar{u}_{\text{bot}}\big) \cdot d_s, \qquad v_{\text{mm},i} = \big(v_{\text{raw},i} - \bar{v}_{\text{bot}}\big) \cdot d_s, \tag{16}$$

where $\bar{u}_{\text{bot}}, \bar{v}_{\text{bot}}$ are computed over $\{i : y_i \le y_{\min} + 0.15\,(y_{\max} - y_{\min})\}$, and d_s is the displacement scale factor (auto-detected so that $\max|u|, \max|v| \approx 10\,\text{mm}$).

3.7 Load Synchronisation and Temporal Derivatives

The UTM and DIC time grids are generally incommensurate. Synchronisation is achieved via piecewise-linear interpolation:

$$P(t_k^{\text{DIC}}) = P(t_i) + \frac{t_k^{\text{DIC}} - t_i}{t_{i+1} - t_i}\big[P(t_{i+1}) - P(t_i)\big], \quad t_i \le t_k^{\text{DIC}} < t_{i+1}. \tag{17}$$

The temporal load-rate $\dot{P}(t)$ is estimated via central finite differences on the smoothed UTM signal:

$$\dot{P}_k = \frac{P_{k+1} - P_{k-1}}{2\Delta t}, \tag{18}$$

and is used directly in the force-rate boundary condition.

4 Input Normalisation and Dimensional Restoration

4.1 Z-Score Normalisation

All five spatio-temporal inputs to the network are independently standardised to zero mean and unit variance:

$$\tilde{q} = \frac{q - \mu_q}{\sigma_q}, \qquad \mu_q = \frac{1}{N}\sum_{i=1}^{N} q_i, \quad \sigma_q = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(q_i - \mu_q)^2}, \tag{19}$$

with the safeguard $\sigma_q \leftarrow 1$ if $\sigma_q < 10^{-8}$. The five normalised quantities are $(\tilde{x}, \tilde{y}, \tilde{t}, \tilde{u}, \tilde{v})$. Statistics (μ_q, σ_q) are stored in a `Normalizer` object and embedded in every saved checkpoint, ensuring inference uses identical scaling.

4.2 Dimensional Restoration via the Chain Rule

The network predicts in normalised coordinates but the PDE requires physical-unit derivatives. Applying the chain rule:

$$\varepsilon_{xx}^{\text{phys}} = \frac{\partial u^{\text{phys}}}{\partial x^{\text{phys}}} = \frac{\partial(\sigma_u \tilde{u} + \mu_u)}{\partial(\sigma_x \tilde{x} + \mu_x)} = \frac{\sigma_u}{\sigma_x}\frac{\partial \tilde{u}}{\partial \tilde{x}}. \tag{20}$$

Analogously:

$$\varepsilon_{yy}^{\text{phys}} = \frac{\sigma_v}{\sigma_y}\frac{\partial \tilde{v}}{\partial \tilde{y}}, \tag{21}$$

$$\gamma_{xy}^{\text{phys}} = \frac{\sigma_u}{\sigma_y}\frac{\partial \tilde{u}}{\partial \tilde{y}} + \frac{\sigma_v}{\sigma_x}\frac{\partial \tilde{v}}{\partial \tilde{x}}, \tag{22}$$

$$\frac{\partial \sigma_{xx}}{\partial x^{\text{phys}}} = \frac{1}{\sigma_x}\frac{\partial \sigma_{xx}}{\partial \tilde{x}}, \tag{23}$$

$$\ddot{u}^{\text{phys}} = \frac{\sigma_u}{\sigma_t^2}\frac{\partial^2 \tilde{u}}{\partial \tilde{t}^2}. \tag{24}$$

All network-computed gradients are therefore multiplied by the appropriate $(\sigma_{\text{output}}/\sigma_{\text{input}})$ ratios before being inserted into the stress or PDE expressions.

5 Neural Network Architecture

5.1 Random Fourier Feature Encoding

Standard MLPs exhibit *spectral bias*: they learn low-frequency functions far more readily than high-frequency ones. To overcome this, raw normalised coordinates are first mapped to a higher-dimensional feature space using the Random Fourier Feature (RFF) embedding.

Definition 5.1 (Random Fourier Feature Map)**.** *Given input* $\in \mathbb{R}^d$ *and a fixed random projection matrix* $\mathbf{B} \in \mathbb{R}^{d\times m}$ *with* $B_{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_B^2)$*, the RFF embedding is:*

$$\gamma(;\mathbf{B}) = \left[\sin\left(2\pi\,{}^{\top}\mathbf{B}\right),\ \cos\left(2\pi\,{}^{\top}\mathbf{B}\right)\right]^{\top} \in \mathbb{R}^{2m}. \tag{25}$$

The resulting inner product approximates the Gaussian kernel: $\langle \gamma(_1), \gamma(_2)\rangle \approx k(_{1,2})$ *with* $k(_{1,2}) = e^{-\|_1 - _2\|^2/(2\sigma_B^2)}$.

The scale σ_B controls the effective bandwidth. Two encoders are used with deliberately different scales, as detailed below.

5.2 Bifurcated DHC-PINN

The network is split into two completely independent branches, reflecting the physical decoupling of kinematic evolution and constitutive behaviour.

DHC_PINN Architecture

Input to both branches: $= (\tilde{x}, \tilde{y}, \tilde{t}) \in \mathbb{R}^3$.

Branch 1 — Kinematic (space × time):

$$\mathbf{B}_{\text{kin}} \in \mathbb{R}^{3\times 64}, \quad B_{ij} \sim \mathcal{N}(0, 0.01^2), \quad \gamma_{\text{kin}}() \in \mathbb{R}^{128}.$$

$$128 \xrightarrow{\text{Linear+SiLU}} 256 \xrightarrow{\text{Linear+SiLU}} 128 \xrightarrow{\text{Linear}} 2 \;\Rightarrow\; (\tilde{u}_{\text{pred}},\, \tilde{v}_{\text{pred}}).$$

Branch 2 — Constitutive (space only, time frozen):

$$_s = (\tilde{x}, \tilde{y}, 0) \in \mathbb{R}^3, \quad \mathbf{B}_{\text{mat}} \in \mathbb{R}^{3\times 16}, \quad B_{ij} \sim \mathcal{N}(0, 0.005^2), \quad \gamma_{\text{mat}}(s) \in \mathbb{R}^{32}.$$

$$32 \xrightarrow{\text{Linear+SiLU}} 128 \xrightarrow{\text{Linear+SiLU}} 64 \xrightarrow{\text{Linear+SiLU}} 32 \xrightarrow{\text{Linear}} 1 \;\Rightarrow\; E_{\text{raw}} \in \mathbb{R}.$$

Remark 5.1. *The material branch receives $t = 0$ explicitly, enforcing the physically motivated assumption that the elastic modulus of a solid specimen does not vary in time. Only the spatial heterogeneity $E(x, y)$ induced by the lattice microstructure is to be recovered. The kinematic branch uses $\sigma_B = 0.01$ (fine spatial/temporal features), while the constitutive branch uses $\sigma_B = 0.005$ (smoother, lower-frequency modulus field). This spectral hierarchy acts as an implicit regulariser.*

5.3 Positivity Enforcement of the Elastic Modulus

The elastic modulus must be positive. A `softplus` offset construction guarantees $E(x, y) > 0$:

$$E(x, y) = \Big[\underbrace{\ln\big(1 + e^{E_{\text{raw}}}\big)}_{\text{softplus}} + 0.5\Big] \times E_{\text{ref}}, \tag{26}$$

so $E(x, y) \geq 0.5\, E_{\text{ref}} > 0$ for all inputs. The shear modulus is then computed analytically from eq. (7).

5.4 Activation Function: SiLU

All hidden layers use the Sigmoid Linear Unit:

$$\text{SiLU}(z) = z \cdot \sigma(z) = \frac{z}{1 + e^{-z}}, \qquad \sigma(z) = \frac{1}{1 + e^{-z}}. \tag{27}$$

SiLU is smooth (C^∞), unbounded above, and has non-zero gradient everywhere, which is important for the second-order autograd derivatives required by the PDE residual.

5.5 Weight Initialisation: Xavier Uniform

All linear layers are initialised with Xavier uniform initialisation:

$$W_{ij} \sim \mathcal{U}\left[-\sqrt{\tfrac{6}{n_{\text{in}}+n_{\text{out}}}},\ +\sqrt{\tfrac{6}{n_{\text{in}}+n_{\text{out}}}}\right], \qquad b_i = 0.01. \tag{28}$$

This ensures that the variance of activations is approximately preserved across layers at initialisation, accelerating early training convergence.

6 The DHC Collocation Cloud

6.1 Sobol Quasi-Random Sequences

The collocation points must cover $\Omega \times [t_s, t_e]$ as uniformly as possible with $N_c = 10{,}000$ samples. Pseudo-random (Monte Carlo) sampling achieves $O(N^{-1/2})$ discrepancy, leaving large empty regions even for large N. The Sobol sequence achieves $O((\log N)^d/N)$ discrepancy in d dimensions — significantly better space-filling coverage.

Definition 6.1 (Discrepancy)**.** *For a point set* $\{z_i\}_{i=1}^N \subset [0,1]^d$*, the* star discrepancy *is*

$$D_N^* = \sup_{J \subseteq [0,1]^d} \left| \frac{\#\{i : z_i \in J\}}{N} - \text{Vol}(J) \right|.$$

Sobol sequences satisfy $D_N^* = O\left(\frac{(\log N)^d}{N}\right)$*, versus* $O(N^{-1/2})$ *for pseudo-random.*

6.2 Construction of the Collocation Cloud

A 3-D scrambled Sobol engine draws N_c samples from $[0,1]^3$, which are then affinely mapped to the physical domain:

$$x_c^{(i)} = p_1^{(i)} \cdot (W - 0) + 0, \tag{29}$$

$$y_c^{(i)} = p_2^{(i)} \cdot (H - 0) + 0, \tag{30}$$

$$t_c^{(i)} = p_3^{(i)} \cdot (t_e - t_s) + t_s, \tag{31}$$

where $(p_1^{(i)}, p_2^{(i)}, p_3^{(i)}) \in [0,1]^3$ is the i^{th} Sobol point. Each coordinate is then normalised using the pre-computed (μ_q, σ_q) statistics before being passed to the network.

The collocation cloud is *resampled* every 100 epochs to avoid overfitting to a fixed set of collocation points. This dynamic resampling improves the spatial coverage over the full training run.

Remark 6.1. *Collocation points carry* ***no displacement labels****. They exist solely to evaluate and penalise violations of the PDE. This is the defining feature of DHC and the mechanism by which physically implausible extrapolation is suppressed.*

7 Loss Functional

The complete DHC loss is the sum of seven terms grouped into two streams. Let $\mathcal{D} = \{(\mathbf{x}_i, t_i)\}_{i=1}^N$ denote the DIC anchor dataset and $\mathcal{C} = \{(x_c^{(j)}, y_c^{(j)}, t_c^{(j)})\}_{j=1}^{N_c}$ the Sobol collocation cloud.

7.1 Stream A: Data Residuals at DIC Anchors

7.1.1 Displacement Data Fidelity

The primary supervised objective fits the network's displacement predictions to the DIC measurements in normalised space:

$$\mathcal{L}_{\text{data}}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\Big[(\tilde{u}_{\text{pred},i} - \tilde{u}_i^{\text{DIC}})^2 + (\tilde{v}_{\text{pred},i} - \tilde{v}_i^{\text{DIC}})^2\Big]. \tag{32}$$

7.1.2 Strain Consistency with Noise-Floor Threshold

The DIC additionally provides direct strain measurements. The strain loss penalises discrepancy only above a noise floor $\epsilon_s = 10^{-3}$:

$$\mathcal{L}_{\text{strain}}(\boldsymbol{\theta}) = \text{ReLU}\left(\frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\left[(\varepsilon_{xx,i}^{\text{pred}} - \varepsilon_{xx,i}^{\text{DIC}})^2 + (\varepsilon_{yy,i}^{\text{pred}} - \varepsilon_{yy,i}^{\text{DIC}})^2 + (\gamma_{xy,i}^{\text{pred}} - \gamma_{xy,i}^{\text{DIC}})^2\right] - \epsilon_s\right), \tag{33}$$

where $\text{ReLU}(z) = \max(0, z)$ ensures that sub-noise-floor agreement is not further penalised. Predicted strains are obtained via eq. (20).

7.1.3 Force-Rate Boundary Condition

Global equilibrium requires that the integral of the axial stress over the cross section equals the applied load:

$$\int_A \sigma_{yy}\,\mathrm{d}A \approx \bar{\sigma}_{yy}\cdot A = P(t). \tag{34}$$

Rather than matching the total load directly (which introduces numerical integration), the *rate* of this relation is enforced pointwise:

$$\mathcal{L}_{\text{force}}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\left(\frac{\partial\sigma_{yy,i}}{\partial t}\cdot A - \dot{P}_i\right)^2, \tag{35}$$

where $\partial\sigma_{yy}/\partial t$ is computed via automatic differentiation and then rescaled by $1/\sigma_t$.

7.1.4 Dirichlet Boundary Condition: Clamped Grip

At the bottom grip Γ_{bot} ($y \le y_{\min} + 0.15\,\Delta y$), both displacement components must vanish in physical space:

$$\mathcal{L}_{\text{Dirichlet}}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}_{\text{bot}}|}\sum_{i\in\mathcal{D}_{\text{bot}}}\Big[(u_i^{\text{phys}})^2 + (v_i^{\text{phys}})^2\Big]. \tag{36}$$

7.1.5 Neumann Boundary Condition: Applied Traction

At the top grip Γ_{top} ($y \ge y_{\max} - 0.15\,\Delta y$), the normal stress must match the applied traction P/A:

$$\mathcal{L}_{\text{Neumann}}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{D}_{\text{top}}|}\sum_{i\in\mathcal{D}_{\text{top}}}\left(\sigma_{yy,i} - \frac{P_i}{A}\right)^2. \tag{37}$$

7.1.6 Tikhonov Smoothness Regularisation of the Modulus Field

To prevent the constitutive branch from fitting noise in the DIC anchor displacements as spurious modulus heterogeneity, a Tikhonov (gradient-norm) penalty is applied to the modulus field:

$$\mathcal{L}_{\text{smooth}}(\boldsymbol{\theta}) = \frac{1}{||}\sum_{i\in} \left[\left(\frac{1}{\sigma_x}\frac{\partial E_i}{\partial \tilde{x}} \right)^2 + \left(\frac{1}{\sigma_y}\frac{\partial E_i}{\partial \tilde{y}} \right)^2 \right]. \tag{38}$$

This is equivalent to a Sobolev H^1 semi-norm penalty on E, biasing the solution towards the spatially smoothest field consistent with the data.

7.2 Stream B: PDE Residual at Sobol Collocation Points

This is the central DHC loss. At each collocation point $(x_c^{(j)}, y_c^{(j)}, t_c^{(j)}) \in \mathcal{C}$ — which carries *no displacement data* — the full Cauchy momentum residual is evaluated:

$$R_x^{(j)} = \frac{\partial \sigma_{xx}^{(j)}}{\partial x} + \frac{\partial \tau_{xy}^{(j)}}{\partial y} - \rho\, \ddot{u}^{(j)}, \tag{39}$$

$$R_y^{(j)} = \frac{\partial \tau_{xy}^{(j)}}{\partial x} + \frac{\partial \sigma_{yy}^{(j)}}{\partial y} - \rho\, \ddot{v}^{(j)}, \tag{40}$$

and the loss is:

$$\mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{C}_{\text{batch}}|} \sum_{j\in\mathcal{C}_{\text{batch}}} \left[(R_x^{(j)})^2 + (R_y^{(j)})^2 \right]. \tag{41}$$

All stress divergence terms are computed via automatic differentiation in normalised coordinates and rescaled by the appropriate σ factors. When $\rho = 0$ (quasi-static), the inertia terms $\rho\ddot{u}, \rho\ddot{v}$ are dropped, so the second-order temporal autodiff graph is not constructed (computational saving).

8 Automatic Differentiation

8.1 Principle

All spatial and temporal derivatives required by the strain, stress-divergence, and inertia expressions are computed by *reverse-mode automatic differentiation* through the full computational graph of the network (including the Fourier encoding). For a scalar output f and leaf variable $\tilde{z}$:

$$\frac{\partial f}{\partial \tilde{z}} = \text{PyTorch autograd with } \texttt{create\_graph=True}, \tag{42}$$

where `create_graph=True` retains the graph of the first derivative to enable computation of second derivatives (for $\ddot{u}, \ddot{v}$). If a gradient is disconnected (e.g. E from t), the function returns $\mathbf{0}$ rather than raising an error (`allow_unused=True`).

8.2 Full Derivative Table in Physical Coordinates

Physical quantity	Computation in normalised coords
ε_{xx}	$\frac{\sigma_u}{\sigma_x}\frac{\partial \tilde{u}}{\partial \tilde{x}}$
ε_{yy}	$\frac{\sigma_v}{\sigma_y}\frac{\partial \tilde{v}}{\partial \tilde{y}}$
γ_{xy}	$\frac{\sigma_u}{\sigma_y}\frac{\partial \tilde{u}}{\partial \tilde{y}} + \frac{\sigma_v}{\sigma_x}\frac{\partial \tilde{v}}{\partial \tilde{x}}$
$\frac{\partial \sigma_{xx}}{\partial x}$	$\frac{1}{\sigma_x}\frac{\partial \sigma_{xx}}{\partial \tilde{x}}$
$\frac{\partial \sigma_{yy}}{\partial t}$	$\frac{1}{\sigma_t}\frac{\partial \sigma_{yy}}{\partial \tilde{t}}$
$\ddot{u}$	$\frac{\sigma_u}{\sigma_t^2}\frac{\partial^2 \tilde{u}}{\partial \tilde{t}^2}$
$\frac{\partial E}{\partial x}$	$\frac{1}{\sigma_x}\frac{\partial E}{\partial \tilde{x}}$

9 Adaptive Lagrangian Optimisation

9.1 Multi-Objective Challenge and Scale Separation

The seven loss terms have wildly different natural scales. For example: the PDE residual at physical stress divergence magnitudes can be $O(10^3\,\mathrm{MPa/mm})$, while the normalised displacement error is $O(1)$. Fixed scalar weights therefore require fragile, problem-specific tuning. The solution is to *learn* the weights.

9.2 Constrained Optimisation Formulation

We frame training as: find network weights $\boldsymbol{\theta}$ that minimise displacement data loss subject to the physics terms being zero:

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) \quad \text{subject to} \quad \mathcal{L}_k(\boldsymbol{\theta}) = 0, \quad k \in \{\text{PDE}, \text{force}, \text{Diric}, \text{Neum}, \text{strain}, \text{smooth}\}. \tag{43}$$

9.3 Augmented Lagrangian Relaxation

The hard constraints in eq. (43) are relaxed via Lagrange multipliers $\lambda_k > 0$, forming the augmented Lagrangian:

$$\mathcal{L}_{\text{Lag}}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \underbrace{\lambda_{\text{data}}\,\mathcal{L}_{\text{data}}}_{\text{fixed, }=1} + \lambda_{\text{PDE}}\,\mathcal{L}_{\text{PDE}} + \lambda_{\text{force}}\,\mathcal{L}_{\text{force}} + 10\lambda_{\text{Diric}}\,\mathcal{L}_{\text{Diric}} + \lambda_{\text{Neum}}\,\mathcal{L}_{\text{Neum}} + \lambda_{\text{strain}}\,\mathcal{L}_{\text{strain}} + \lambda_{\text{smooth}}\,\mathcal{L}_{\text{sm}} \tag{44}$$

The factor 10 on the Dirichlet term encodes a prior that the rigid-body boundary condition is more reliably satisfied than the other soft constraints.

9.4 Saddle-Point Problem

The optimal solution satisfies the *min–max* saddle-point condition:

$$(\boldsymbol{\theta}^*, \boldsymbol{\lambda}^*) = \arg\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\lambda} \geq 0} \ \mathcal{L}_{\text{Lag}}(\boldsymbol{\theta}, \boldsymbol{\lambda}). \tag{45}$$

The intuition is: if a constraint $\mathcal{L}_k > 0$ is violated, maximising over λ_k increases it, making the Lagrangian larger, which forces the minimising $\boldsymbol{\theta}$ to reduce $\mathcal{L}_k$ further.

9.5 Log-Space Parameterisation of Multipliers

To guarantee $\lambda_k > 0$ and allow unconstrained gradient ascent, each multiplier is parameterised in log space:

$$\lambda_k = \min\big(e^{\ell_k},\ \Lambda_{\max}\big), \qquad \ell_k \in \mathbb{R}, \quad \Lambda_{\max} = 10^5. \tag{46}$$

The six log parameters and their initial values are:

Parameter	$\ell_k^{(0)}$	$\lambda_k^{(0)} \approx$	Physical meaning
ℓ_{PDE}	2.3	10	Momentum balance
ℓ_{force}	7.0	1097	Force-rate BC
ℓ_{Diric}	6.9	992	Bottom grip ($\times$10 extra)
ℓ_{Neum}	6.9	992	Top traction BC
ℓ_{strain}	4.6	100	Strain consistency
ℓ_{smooth}	-4.6	0.01	Modulus smoothness

Remark 9.1. *The very high initial values of λ_{force}, λ_{Diric}, and λ_{Neum} reflect the fact that these boundary conditions are inviolable physical facts (the grip* is *clamped; the machine load* is *applied), so the network is strongly penalised from the very first epoch if it violates them. In contrast, λ_{smooth} starts small to allow the modulus field to fit the data before regularisation kicks in.*

9.6 Dual Gradient Ascent with Adam

The optimisation is solved by two simultaneous Adam updates:

DHC-PINN Training Algorithm

Inputs: DIC dataloader $\mathcal{D}$; collocation cloud $\mathcal{C}$ of size $N_c = 10{,}000$; epochs $T = 10{,}000$; $\alpha_\theta = 10^{-3}$; $\alpha_\lambda = 10^{-3}$.

Every $N_r = 100$ epochs: Resample $\mathcal{C}$ from new Sobol draw.

For each epoch s and data mini-batch :

Step 1. Enable autograd on $(\tilde{x},\tilde{y},\tilde{t})$, (x_c, y_c, t_c).

Step 2. Stream A — Forward pass on DIC anchors: Compute $(\tilde{u}_{\text{pred}}, \tilde{v}_{\text{pred}}, E, G)$ at ; evaluate all six anchor-side residuals $(\mathcal{L}_{\text{data}}, \mathcal{L}_{\text{strain}}, \mathcal{L}_{\text{force}}, \mathcal{L}_{\text{Diric}}, \mathcal{L}_{\text{Neum}}, \mathcal{L}_{\text{smooth}})$.

Step 3. Stream B — Forward pass on Sobol collocation: Compute predictions at a collocation mini-batch $\mathcal{C}$; evaluate $\mathcal{L}_{\text{PDE}}$ via eq. (41).

Step 4. Retrieve multipliers: $(\lambda_{\text{PDE}}, \ldots) \leftarrow \exp(\boldsymbol{\ell})$ via eq. (46).

Step 5. Assemble Lagrangian: $\mathcal{L}_{\text{Lag}} \leftarrow$ eq. (44).

Step 6. Backward pass: Compute $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{Lag}}$ and $\nabla_{\boldsymbol{\ell}}\mathcal{L}_{\text{Lag}}$.

Step 7. Gradient clipping: $\hat{\nabla}_{\boldsymbol{\theta}} \leftarrow \nabla_{\boldsymbol{\theta}} / \max(1, \|\nabla_{\boldsymbol{\theta}}\|_2)$.

Step 8. Primal descent (Adam, every batch):
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha_\theta\, \widehat{\mathbf{m}}_{\boldsymbol{\theta}^{(s)}} / (\sqrt{\hat{\mathbf{v}}_{\boldsymbol{\theta}^{(s)}}} + \varepsilon)$.

Step 9. Dual ascent (Adam `maximize=True`, every 5 batches):
$\boldsymbol{\ell} \leftarrow \boldsymbol{\ell} + \alpha_\lambda\, \widehat{\mathbf{m}}_{\boldsymbol{\ell}}^{(s)} / (\sqrt{\hat{\mathbf{v}}_{\boldsymbol{\ell}}^{(s)}} + \varepsilon)$.

LR Scheduler: `ReduceLROnPlateau` — halve α_θ if $\mathcal{L}_{\text{Lag}}$ does not decrease for 150 consecutive epochs.

9.6.1 Adam Update Rule

The Adam update for parameter vector $\boldsymbol{\phi}$ at global step s is:

$$\mathbf{m}^{(s)} = \beta_1 \mathbf{m}^{(s-1)} + (1-\beta_1)\,\nabla\mathcal{L}, \tag{47}$$

$$\mathbf{v}^{(s)} = \beta_2 \mathbf{v}^{(s-1)} + (1-\beta_2)\,(\nabla\mathcal{L})^{\odot 2}, \tag{48}$$

$$\hat{\mathbf{m}}^{(s)} = \mathbf{m}^{(s)} / (1-\beta_1^s), \tag{49}$$

$$\hat{\mathbf{v}}^{(s)} = \mathbf{v}^{(s)} / (1-\beta_2^s), \tag{50}$$

$$\boldsymbol{\phi}^{(s+1)} = \boldsymbol{\phi}^{(s)} \mp \frac{\alpha\, \hat{\mathbf{m}}^{(s)}}{\sqrt{\hat{\mathbf{v}}^{(s)}} + \varepsilon_{\text{Adam}}}, \tag{51}$$

with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon_{\text{Adam}} = 10^{-8}$. The minus sign applies to primal descent ($\boldsymbol{\theta}$); the plus sign applies to dual ascent ($\boldsymbol{\ell}$), implemented via `maximize=True`.

9.6.2 The Staggered Multiplier Update

Multiplier updates are executed only every 5 data batches. This staggering is deliberate: it allows the primal variables $\boldsymbol{\theta}$ to partially adapt to the current multiplier values before they are revised, preventing the oscillation characteristic of simultaneous gradient-based saddle-point solvers.

9.7 ReduceLROnPlateau Scheduler

After every epoch, the scheduler monitors $\mathcal{L}_{\text{Lag}}$. If no improvement is observed for $p = 150$ epochs, the learning rate is reduced:

$$\alpha_\theta \leftarrow f \cdot \alpha_\theta, \qquad f = 0.5. \tag{52}$$

This is equivalent to an exponential cooling schedule triggered by stagnation, and helps the optimisation escape shallow plateaus near the end of training.

10 Post-Training Analysis

10.1 Physical Plausibility Check

After training, 500 random interior probe points are sampled uniformly from Ω at the mean training time $\bar{t}$. The predicted elastic modulus $E(x, y)$ is evaluated and a warning

is issued if $E \notin [50,\ 50{,}000]$ MPa for any probe. This self-diagnostic detects calibration errors (e.g. wrong DISP_SCALE) before the expensive inference pass.

10.2 Derived Physical Fields

Given converged $\boldsymbol{\theta}^*$, the following fields are reconstructed on a uniform $N_g \times N_g$ ($N_g = 300$–400) spatial grid at query times $t^* \in \{q, q + \frac{1}{2}\sigma_t, q - \frac{1}{2}\sigma_t\}$ where q is the midpoint of the training interval:

Strain fields. Computed via autograd through the network as in section 4.2:

$$\varepsilon_{xx}(x, y), \quad \varepsilon_{yy}(x, y), \quad \gamma_{xy}(x, y). \tag{53}$$

Stress fields. From the constitutive law eq. (6):

$$\sigma_{xx} = \frac{E}{1-\nu^2}(\varepsilon_{xx} + \nu\varepsilon_{yy}), \quad \sigma_{yy} = \frac{E}{1-\nu^2}(\varepsilon_{yy} + \nu\varepsilon_{xx}), \quad \tau_{xy} = G\,\gamma_{xy}. \tag{54}$$

Von Mises effective stress (plane stress):

$$\sigma_{\text{VM}} = \sqrt{\sigma_{xx}^2 - \sigma_{xx}\sigma_{yy} + \sigma_{yy}^2 + 3\tau_{xy}^2}. \tag{55}$$

Constitutive field maps: $E(x, y)$, $G(x, y)$ from the material branch.

Each contour map overlays the DIC anchor positions as scatter markers, providing a direct visual comparison between the sparse measurement locations and the network's dense field reconstruction.

11 Complete Mathematical Pipeline

End-to-End DHC-PINN Mathematical Pipeline

$$\underbrace{(x, y, t)_{\text{phys}}}_{\substack{\text{Raw DIC +} \\ \text{UTM data}}} \xrightarrow{\text{calibration, SG filter, Hookean isolation}} \underbrace{(\tilde{x}, \tilde{y}, \tilde{t}, \tilde{u}, \tilde{v})}_{\text{normalised inputs}}$$

$$\underbrace{\gamma_{\text{kin}}(\tilde{x}, \tilde{y}, \tilde{t}) \in \mathbb{R}^{128}}_{\sigma_B=0.01,\ m_{\text{kin}}=64} \qquad \underbrace{\gamma_{\text{mat}}(\tilde{x}, \tilde{y}, 0) \in \mathbb{R}^{32}}_{\sigma_B=0.005,\ m_{\text{mat}}=16}$$

$$\underbrace{(\tilde{u}_{\text{pred}},\ \tilde{v}_{\text{pred}})}_{\text{kinematics}} \qquad \underbrace{E(x, y),\ G(x, y)}_{\text{constitutive fields}}$$

$$\underbrace{\varepsilon_{ij},\ \sigma_{ij},\ \nabla \cdot \boldsymbol{\sigma},\ \ddot{\mathbf{u}}}_{\text{via autograd + dimensional rescaling}}$$

$$\mathcal{L}_{\text{Lag}} = \mathcal{L}_{\text{data}} + \lambda_{\text{PDE}}\mathcal{L}_{\text{PDE}} + \lambda_f\mathcal{L}_f + 10\lambda_d\mathcal{L}_d + \lambda_n\mathcal{L}_n + \lambda_s\mathcal{L}_s + \lambda_{\text{sm}}\mathcal{L}_{\text{sm}}$$

$$\Big\downarrow \text{ Adam primal descent on } \boldsymbol{\theta} \;+\; \text{Adam dual ascent on } \boldsymbol{\ell}$$

$$\boldsymbol{\theta}^*,\ \boldsymbol{\lambda}^* \implies E^*(x,y),\ G^*(x,y),\ \sigma^*_{\text{VM}}(x,y),\ (u^*,v^*)(x,y,t)$$

11.1 Hyperparameter Reference Table

Symbol	Value	Description	Location in code
s_f	15	DIC frame stride	`STRIDE_FACTOR`
T	10,000	Training epochs	`EPOCHS`
$B_{\mathcal{D}}$	16,384	DIC mini-batch	`BATCH_SIZE_DATA`
N_c	10,000	Sobol colloc points	`N_COLLOC`
$B_{\mathcal{C}}$	16,384	Colloc mini-batch	`BATCH_SIZE_COLLOC`
N_r	100	Colloc resample freq.	`RESAMPLE_COLLOC_EVERY`
α_θ	10^{-3}	Primal LR	`LR_THETA`
α_λ	10^{-3}	Dual LR	`LR_LAMBDA`
m_{kin}	64	Fourier features (kin)	`FOURIER_KIN`
m_{mat}	32	Fourier features (mat)	`FOURIER_MAT`
σ_B^{kin}	0.01	Fourier scale (kin)	`DHC_PINN.__init__`
σ_B^{mat}	0.005	Fourier scale (mat)	`DHC_PINN.__init__`
ϵ_s	10^{-3}	Strain noise floor	`EPS_STRAIN`
$\Lambda_{\max}$	10^5	Multiplier ceiling	`LAMBDA_CEILING`
f	0.5	LR reduction factor	`scheduler`
p	150	LR patience (epochs)	`scheduler`
$R^2_{\min}$	0.999	Hookean threshold	`R2_THRESHOLD`
w	5	Hookean window width	`HOOKEAN_WINDOW`

Appendix: Notation Reference

Symbol	Meaning
$\Omega \subset \mathbb{R}^2$	Spatial domain
W, H	Specimen width and height (mm)
A	Cross-sectional area (mm^2)
u, v	Displacement components in x- and y-directions (mm)
$\varepsilon_{xx}, \varepsilon_{yy}$	Normal strains (dimensionless)
γ_{xy}	Engineering shear strain (dimensionless)
σ_{xx}, σ_{yy}	Normal stresses (MPa)
τ_{xy}	Shear stress (MPa)
$E(x, y)$	Spatially-varying Young's modulus (MPa)
$G(x, y)$	Spatially-varying shear modulus (MPa)
ν	Poisson's ratio
ρ	Mass density (t/mm^3)
$P(t)$	Applied load (N) from UTM
$\dot{P}$	Load rate (N/s)
E_{ref}	Reference elastic modulus for softplus anchor (MPa)
$\boldsymbol{\theta}$	Neural network trainable parameters
$\boldsymbol{\lambda} = (\lambda_k)$	Lagrange multipliers (adaptively learned)
$\boldsymbol{\ell} = (\ell_k)$	Log-space multiplier parameters
$\gamma(\cdot; \mathbf{B})$	Random Fourier Feature encoding
$\mathbf{B}$	Fixed random projection matrix
σ_B	Fourier feature scale (std dev of $\mathbf{B}$ entries)
$\text{SiLU}(\cdot)$	Sigmoid linear unit activation
$\text{softplus}(\cdot)$	$\ln(1 + e^z)$ positivity-enforcing function
$\mathcal{L}_k$	Individual loss terms
$\mathcal{L}_{\text{Lag}}$	Adaptive Lagrangian objective
$\mathcal{D}$	DIC anchor dataset
$\mathcal{C}$	Sobol collocation cloud
$\tilde{(\cdot)}$	Normalised (dimensionless) quantity
μ_q, σ_q	Mean and standard deviation of channel q
	Data mini-batch
N_c	Number of collocation points
s_f	Temporal stride factor