APPLICATIONS OF
MODELLING AND SIMULATION
http://arqiipubl.com/ams eISSN 2600-8084 VOL 10, 2026, 238–252

Physics-Informed Neural Networks for Steady Viscoelastic Flows in Fiber Spinning and Extrusion Film Casting

Melwina Albuquerque[1], Renu Dhadwal[2,*], Jayaraman Valadi[2,*] and Rohan Lodhi[2]

[1]Center for Interdisciplinary Artificial Intelligence, FLAME University, Pune, India
[2]Department of Computing and Data Science, FLAME University, Pune, India

[*]Corresponding author: renu.dhadwal@flame.edu.in, jayaraman.vk@flame.edu.in

Submitted 28 April 2026; Revised 31 July 2026; Accepted 17 August 2026; Available online 10 September 2026.

Abstract: We investigate Physics-Informed Neural Networks (PINNs) for two classical steady-state viscoelastic flow problems in polymer processing: fiber spinning and extrusion film casting. Both problems are governed by coupled nonlinear differential equations with viscoelastic constitutive behavior and serve as useful test cases for assessing PINNs on structured boundary-value problems. For the fiber-spinning model, we employ a modular PINN architecture with separate networks for area, velocity, and stress. For the extrusion film-casting model, we use a monolithic PINN that jointly predicts velocity, geometry, and stress fields while also inferring an unknown stress-related parameter. In both cases, the PINN solutions are validated against numerical reference solutions obtained from conventional integration-based methods. The results show that PINNs can recover the principal steady solution fields with high accuracy for the parameter regimes considered. For fiber spinning, the modular architecture yields strong agreement with the numerical solution across all variables. For extrusion film casting, the baseline PINN captures the kinematic variables well, while stress prediction is more challenging because of stiffness and steep gradients. This difficulty is substantially reduced by using adaptive activation functions, leading to marked improvement in stress-field accuracy. In addition, the unknown model parameter in the film-casting problem is inferred with good agreement relative to the numerical reference value. These results indicate that PINNs provide a flexible mesh-free framework for solving and parameterizing steady viscoelastic flow models, while also highlighting the greater difficulty of accurately resolving stress fields in stiff regimes.

Keywords: Extrusion film casting; Fiber spinning; Physics-informed neural networks; Polymer processing; Viscoelastic flow.

1. INTRODUCTION

Polymer-processing operations such as fiber spinning and extrusion film casting (EFC) are governed by strongly coupled interactions among flow kinematics, geometry, and viscoelastic stress. In fiber spinning, a molten polymer is extruded through a fine orifice and drawn into a filament, whereas in EFC a thin polymer film is extruded through a slit die and stretched under tension before cooling. These processes are important in textile, packaging, and related manufacturing applications, and their performance depends sensitively on the evolution of velocity, geometry, and stress fields along the flow direction.

Even in reduced steady one-dimensional formulations, both processes lead to nonlinear boundary-value problems involving conservation laws and viscoelastic constitutive equations. Their reduced dimensionality does not make them equivalent to conventional initial-value problems. In fiber spinning, the inlet stress is not prescribed directly but must be determined so that the specified outlet draw ratio is satisfied. In EFC, an unknown force parameter must similarly be determined together with the velocity, geometric, and stress fields through an outlet condition. The strong coupling among these variables, together with constitutive nonlinearity, parameter sensitivity, and steep stress gradients, can make the numerical treatment nontrivial despite the apparently simple one-dimensional geometry.

Classical analyses of steady viscoelastic fiber spinning and film casting are well established. For fiber spinning, Maxwell-type and related viscoelastic models have been studied extensively using continuum and numerical approaches, including shooting and continuation methods [1–4]. For extrusion film casting, substantial work has focused on one-dimensional and higher-dimensional viscoelastic models capable of capturing physically important effects such as neck-in, draw resonance, and stress evolution. Barborik and Zatloukal [5] reviewed steady isothermal models based on Maxwell and Phan–Thien–Tanner (PTT) constitutive laws, Bechert [6] examined instability mechanisms using asymptotic analysis, and Shin *et al.* [7] studied transient and steady two-dimensional non-isothermal viscoelastic film casting. Foundational studies by Debbaut *et al.* [8] and Anturkar and Co [9] further clarified the linear and nonlinear stability properties of viscoelastic film flows.

For the reduced steady models considered here, conventional numerical solutions can be obtained using shooting methods. In such methods, the governing equations are integrated repeatedly as an initial-value problem while an unknown inlet or force quantity is adjusted until the prescribed outlet condition is satisfied. These methods are computationally efficient when a reliable initial guess or parameter bracket is available. However, the forward integration and the determination of the shooting quantity remain separate numerical steps.

Physics-Informed Neural Networks (PINNs), introduced by Raissi *et al.* [10], provide an alternative framework for solving differential equations by representing the unknown solution with a neural network trained to satisfy the governing equations and boundary conditions. Derivatives of the network outputs are evaluated through automatic differentiation, allowing the differential-equation residuals to be incorporated directly into the training objective. Unknown physical parameters may also be represented as trainable variables and inferred jointly with the solution fields within the same optimization framework.

PINNs have previously been applied to a range of non-Newtonian and viscoelastic-flow problems. These include nn-PINNs for complex-fluid modelling [11], ViscoelasticNet for stress inference and constitutive-model identification [12], and training strategies intended to improve convergence in difficult or stiff systems [13]. More general developments in PINN optimization and stability include adaptive loss balancing [14], combinations of first- and second-order optimization methods [15, 16], and adaptive activation functions [17].

Despite these developments, the particular steady viscoelastic boundary-value systems considered here have not, to the best of our knowledge, previously been investigated within a PINN framework. The fiber-spinning and EFC models are reduced special cases of viscoelastic flow and are not claimed to be more general than previously studied multidimensional viscoelastic systems. Nevertheless, their reduction to one spatial dimension does not make their numerical structure equivalent to that of a conventional initial-value problem. Both are two-point boundary-value systems in which an unknown inlet or force quantity must be determined so that an outlet draw-ratio condition is satisfied. In the EFC model, this global constraint is combined with algebraic conservation relations, nonlinear PTT stress evolution, and steep stress gradients. These systems therefore provide a distinct benchmark setting for evaluating PINN optimization and parameter inference, even though the constituent PINN techniques are established.

In this paper, we investigate PINN-based solvers for two canonical steady viscoelastic polymer-processing problems: fiber spinning and extrusion film casting. The two case studies serve different purposes. Fiber spinning provides a comparatively compact setting in which the area, velocity, and stress fields are coupled through mass conservation, force balance, and a Maxwell constitutive equation. It is used to compare a modular PINN, consisting of separate variable-specific subnetworks, with monolithic PINNs under matched training conditions and comparable parameter counts.

The EFC problem provides a more demanding setting in which the network must predict six coupled kinematic, geometric, and stress variables while simultaneously estimating the unknown force parameter E. In the conventional shooting formulation, E is adjusted externally until the specified outlet velocity is attained. In the PINN formulation, E is optimized jointly with the solution fields. The EFC problem is also used to assess the accuracy and optimization robustness of a baseline PINN and an adaptive-activation configuration. Adaptive activation functions and the associated training strategies are established techniques; their role here is to assess their effectiveness for the particular EFC boundary-value system rather than to introduce a new general PINN method.

The principal contributions of this work are as follows:

- We formulate and evaluate PINN solvers for the steady Maxwell fiber-spinning and biaxial PTT extrusion-film-casting boundary-value systems.
- We provide a controlled comparison of modular and monolithic PINN representations for the fiber-spinning problem, including a parameter-matched monolithic model that separates the effect of network representation from that of parameter count.
- We formulate the EFC problem as a joint solution-and-parameter-estimation problem in which the unknown force parameter E is learned together with the six coupled solution fields.
- We assess the effects of constitutive-residual scaling, loss-weight choices, training configuration, and random initialization on solution accuracy and optimization robustness.
- We validate the PINN predictions against independently generated shooting-based reference solutions using consistent field-wise and aggregate error metrics.

Overall, the paper evaluates PINNs as an alternative differentiable framework for coupled steady viscoelastic boundary-value problems, particularly when the solution fields and an unknown physical parameter are to be determined within a single optimization framework. The objective is not to suggest that PINNs should replace established shooting methods for rapid forward computation, but to examine their accuracy, robustness, and inverse-modelling capability for these structured polymer-processing problems.

The remainder of this paper is organized as follows. Section 2 presents the general PINN methodology, including the network formulation, loss construction, boundary-condition treatment, and error metrics. Section 3 introduces the governing equations, physical assumptions, parameters, and boundary conditions for the fiber-spinning and EFC models. Section 4 describes the independent numerical reference solutions, the problem-specific PINN formulations, and the corresponding numerical results. Section 5 summarizes the principal findings and outlines directions for future work.

2. PHYSICS-INFORMED NEURAL NETWORK (PINN) METHODOLOGY

Physics-Informed Neural Networks (PINNs) approximate the solution of a differential system by a neural network trained to satisfy the governing equations and boundary conditions. In the present work, the models are steady one-dimensional boundary-value problems posed on an interval, so the network depends on a single spatial coordinate and derivatives are taken with respect to that variable.

Let $\Omega = (0,1)$ and let

$$\mathbf{y} : \Omega \to \mathbb{R}^m, \qquad \mathbf{y}(x) = (y_1(x), \ldots, y_m(x))^{\mathsf{T}},$$

denote the unknown state vector. We write the governing system in residual form as

$$\mathscr{N}\left(x, \mathbf{y}, \partial_x \mathbf{y}; \phi\right) = \mathbf{0}, \qquad x \in (0,1), \tag{1}$$

together with boundary conditions

$$\mathscr{B}\big(\mathbf{y}(0), \mathbf{y}(1)\big) = \mathbf{0}. \tag{2}$$

Here $\phi \in \mathbb{R}^p$ denotes the collection of physical parameters. When some components of ϕ are unknown, they may be treated as trainable variables and inferred jointly with the solution.

We approximate $\mathbf{y}$ by a feed-forward neural network

$$\hat{\mathbf{y}}(x;\theta) : \Omega \to \mathbb{R}^m,$$

with trainable parameters θ. Using smooth activation functions, $\hat{\mathbf{y}}$ is differentiable with respect to x, and automatic differentiation provides the derivatives required to evaluate the residual in Equation (1). Defining the pointwise residual

$$\mathbf{r}(x;\theta,\phi) := \mathscr{N}\left(x, \hat{\mathbf{y}}(x;\theta), \partial_x \hat{\mathbf{y}}(x;\theta); \phi\right),$$

we enforce the governing equations at interior collocation points $\{x_i\}_{i=1}^{N_f} \subset (0,1)$ through the physics loss

$$\mathscr{L}_{\text{phys}}(\theta,\phi) = \frac{1}{N_f} \sum_{i=1}^{N_f} \|\mathbf{r}(x_i;\theta,\phi)\|_2^2 .$$

Boundary conditions are imposed either softly, through penalty terms, or exactly, through output transformations that satisfy selected Dirichlet constraints by construction. In the soft-constraint setting, the boundary loss is written as

$$\mathscr{L}_{\text{bc}}(\theta) = \sum_{k=1}^{K} \lambda_k \,\|\mathscr{B}_k(\hat{\mathbf{y}}(0;\theta), \hat{\mathbf{y}}(1;\theta))\|_2^2,$$

where $\lambda_k > 0$ are penalty weights. In the hard-constraint setting, transformed outputs are used and the corresponding penalty terms are omitted.

The total loss takes the general form

$$\mathscr{L}_{\text{total}}(\theta,\phi) = w_f\,\mathscr{L}_{\text{phys}} + w_b\,\mathscr{L}_{\text{bc}} + \mathscr{L}_{\text{stab}},$$

where $\mathscr{L}_{\text{stab}}$ denotes optional stabilization terms used when needed, such as gradient penalties or problem-specific residual reweighting.

Training is performed by minimizing the relevant loss terms over the collocation points, and the trained models are evaluated on a separate dense grid by comparison with numerical reference solutions. Because the two case studies considered in this paper have different structures and training requirements, architectural choices, optimizer settings, collocation strategies, and loss weights are specified separately in the corresponding problem sections.

Error metrics. To report accuracy consistently across both case studies, we evaluate each trained model on a dense reference grid of N points and use the same four metrics throughout. Let a model have V solution fields, and for field j let $\mathbf{p}_j = (\hat{y}_j(x_1), \ldots, \hat{y}_j(x_N))^{\mathsf{T}}$ and $\mathbf{r}_j = (y_j(x_1), \ldots, y_j(x_N))^{\mathsf{T}}$ denote the predicted and reference values at the grid points. We report:

- **Relative L_2 error (per field):** $\varepsilon_j = \dfrac{\|\mathbf{p}_j - \mathbf{r}_j\|_2}{\|\mathbf{r}_j\|_2}$, the Euclidean norm taken *over grid points*.
- **Mean relative L_2 error:** $\dfrac{1}{V}\displaystyle\sum_{j=1}^{V} \varepsilon_j$, the per-field relative L_2 error averaged *over fields*.
- **Maximum absolute error:** $\displaystyle\max_{1\le j\le V}\ \max_{1\le i\le N} \left|\hat{y}_j(x_i) - y_j(x_i)\right|$, the maximum taken *over fields and grid points*.
- **Boundary MAE:** $\dfrac{1}{B}\displaystyle\sum_{b=1}^{B} \left|\hat{y}_b - y_b^{\text{target}}\right|$, the mean absolute deviation *over the B imposed boundary conditions*. For models with hard output transforms the imposed conditions are satisfied by construction, so this metric is exactly zero.

In addition, we report the coefficient of determination R^2 per field, and a total R^2 defined as the mean of the per-field values. Each scalar is computed per random seed and reported as its mean ± standard deviation over seeds. This fixes the order of aggregation: relative L_2 is a norm over grid points, then a mean over fields, then a mean over seeds; the maximum absolute error is a maximum over fields and points, then a mean over seeds.

3. GOVERNING EQUATIONS AND PHYSICAL MODELS

In this section, we introduce the two processes of fiber spinning and extrusion film casting and describe their governing equations.

3.1 Fiber Spinning

We consider a simplified steady one-dimensional model for viscoelastic fiber spinning with a single axial stress component. The formulation follows Dhadwal *et al.* [18] and is written in dimensionless form. The unknown fields are the cross-sectional area $a(x)$, the axial velocity $v(x)$, and the axial stress $\tau(x)$, all depending on the axial coordinate $x \in [0,1]$. These quantities are cross-sectionally averaged and describe an extensional flow of an incompressible Maxwell fluid.

The governing equations are

$$a\frac{dv}{dx} + v\frac{da}{dx} = 0, \tag{3}$$

$$a\frac{d\tau}{dx} + \tau\frac{da}{dx} = 0, \tag{4}$$

$$v\frac{d\tau}{dx} - 2\left(\tau + \frac{1}{De}\right)\frac{dv}{dx} + \frac{\tau}{De} = 0. \tag{5}$$

Equation (3) represents conservation of mass, Equation (4) expresses axial force balance in the absence of inertia, gravity, and surface tension, and Equation (5) is the steady Maxwell constitutive equation. Here De denotes the Deborah number, which measures the ratio of the material relaxation time to the characteristic process time.

The system is supplemented by the boundary conditions

$$a(0) = 1, \tag{6}$$

$$v(0) = 1, \tag{7}$$

$$v(1) = D, \tag{8}$$

where D is the draw ratio, defined as the ratio of outlet velocity to inlet velocity. Physically, the draw ratio quantifies the total imposed stretch and strongly influences the final fiber thickness and stress development.

In conventional numerical treatments, this boundary-value problem is typically solved by a shooting method in which the unknown inlet stress $\tau(0)$ is treated as the shooting parameter and is adjusted until the outlet condition $v(1) = D$ is satisfied.

3.2 Extrusion Film Casting

We consider a steady one-dimensional model of biaxial extrusion film casting for an incompressible viscoelastic polymer melt. The formulation involves six coupled dimensionless variables: the axial velocity $u(x)$, the film half-width $L(x)$, the film half-thickness $e(x)$, and the normal stress components $\tau_{xx}(x)$, $\tau_{yy}(x)$, and $\tau_{zz}(x)$, all depending on the axial coordinate $x \in [0,1]$. The governing equations are taken from Dhadwal *et al.* [19].

The model is

$$eLu = 1, \tag{9}$$

$$\tau_{xx} - \tau_{zz} = u, \tag{10}$$

$$\frac{dL}{dx} + A\sqrt{\frac{\tau_{yy} - \tau_{zz}}{\tau_{xx} - \tau_{zz}}} = 0, \tag{11}$$

$$De\, u\frac{d\tau_{xx}}{dx} + \frac{du}{dx}\left(-2De\,\tau_{xx} + 4De\,\psi\,\tau_{xx} - 2E\right) + \tau_{xx}\exp\left(\frac{De\,\varepsilon}{E}\left(\tau_{xx} + \tau_{yy} + \tau_{zz}\right)\right) = 0, \tag{12}$$

$$De\, u\frac{d\tau_{yy}}{dx} + \frac{dL}{dx}\left(-2De\,\tau_{yy}\frac{u}{L} + 4De\,\psi\frac{\tau_{yy}}{L} - 2E\frac{u}{L}\right) + \tau_{yy}\exp\left(\frac{De\,\varepsilon}{E}\left(\tau_{xx} + \tau_{yy} + \tau_{zz}\right)\right) = 0, \tag{13}$$

$$De\, u\frac{d\tau_{zz}}{dx} + \frac{de}{dx}\left(-2De\,\tau_{zz}\frac{u}{e} + 4De\,\psi\frac{\tau_{zz}}{e} - 2E\frac{u}{e}\right) + \tau_{zz}\exp\left(\frac{De\,\varepsilon}{E}\left(\tau_{xx} + \tau_{yy} + \tau_{zz}\right)\right) = 0. \tag{14}$$

Equation (9) expresses conservation of mass, Equation (10) is the reduced axial momentum relation, Equation (11) governs the geometric evolution of the film width, and Equations (12)–(14) are the viscoelastic stress-evolution equations corresponding to the Phan–Thien–Tanner (PTT) constitutive model. Here De is the Deborah number, while ε and ψ are PTT model parameters. The parameter A controls the geometric evolution, and E is an unknown dimensionless force parameter that appears in the stress equations.

The inlet boundary conditions are

$$u(0) = 1, \qquad L(0) = 1, \qquad e(0) = 1, \qquad \tau_{xx}(0) = 1, \qquad \tau_{yy}(0) = 0.5, \qquad \tau_{zz}(0) = 0, \tag{15}$$

and the outlet condition is

$$u(1) = D, \tag{16}$$

where D is the draw ratio. In conventional numerical solution, the unknown parameter E is determined as a shooting parameter so that the outlet condition Equation (16) is satisfied.

4. NUMERICAL RESULTS AND DISCUSSION

4.1 Numerical Reference Solutions

4.1.1 Fiber-spinning Reference Solution

The numerical reference solution for the fiber-spinning problem was generated independently of the PINN training procedure using a shooting method. The unknown inlet stress,

$$\tau(0) = \tau_0,$$

was treated as the shooting parameter. For each trial value of τ_0, equations (3)–(5) were solved as an initial-value problem on $x \in [0,1]$, subject to

$$a(0) = 1, \qquad v(0) = 1, \qquad \tau(0) = \tau_0.$$

The initial-value problem was integrated in MATLAB using the `ode23tb` solver. MATLAB's default error-control and step-size settings were used; in particular, the default relative and absolute tolerances were $\mathrm{RelTol} = 10^{-3}$ and $\mathrm{AbsTol} = 10^{-6}$, respectively.

The shooting residual was defined by

$$F_{\mathrm{FS}}(\tau_0) = v(1;\tau_0) - D,$$

where D is the prescribed draw ratio. Successive approximations to τ_0 were generated using the secant method. The shooting iteration was terminated when

$$|F_{\mathrm{FS}}(\tau_0)| = |v(1;\tau_0) - D| < 10^{-3}.$$

The converged solution was evaluated at 100 uniformly spaced output points on $x \in [0,1]$ and was used as the reference solution for the post-training evaluation of the PINN predictions. No values from the reference solution were included in the PINN loss function or otherwise used during PINN training. The physical parameters used are given in Table 1 and the numerical settings are given in Table 2.

4.1.2 Extrusion-film-casting Reference Solution

The numerical reference solution for the extrusion-film-casting problem was generated independently of the PINN training procedure using a shooting method. The unknown force parameter E was treated as the shooting parameter. For each trial value of E, the governing system was solved as an initial-value problem on $x \in [0,1]$, subject to the inlet conditions

$$u(0) = 1, \qquad L(0) = 1, \qquad e(0) = 1,$$

$$\tau_{xx}(0) = 1, \qquad \tau_{yy}(0) = 0.5, \qquad \tau_{zz}(0) = 0.$$

The initial-value problem was integrated in MATLAB using the `ode15s` solver. MATLAB's default integration settings were used. For each trial value of E, the discrepancy between the computed outlet velocity and the prescribed draw ratio was measured using the objective function

$$J(E) = |u(1;E) - D|.$$

The value of E was determined by minimizing $J(E)$ over the bounded interval

$$E \in [0.1, 1]$$

using the MATLAB routine `fminbnd`. The default optimization settings were used; in particular, the termination tolerance on E was $\mathrm{TolX} = 10^{-4}$, and the maximum numbers of iterations and objective-function evaluations were both 500.

The converged reference value was

$$E = 0.1770.$$

The corresponding solution was evaluated at 1000 uniformly spaced output points over $x \in [0,1]$ and was used for the post-training evaluation of the PINN predictions. No values from the numerical reference solution were included in the PINN loss function or otherwise used during PINN training. The physical parameters used are given in Table 1. The numerical settings are given in Table 2.

4.2 PINN results

The network architectures and training settings used for the fiber-spinning, baseline EFC, and adaptive-activation EFC models are summarized in Table 3.

Table 1. Physical parameters, boundary conditions, and unknown shooting quantities for the fiber-spinning and extrusion-film-casting problems.

Process	Constitutive model	Parameter values	Boundary conditions	Unknown shooting quantity
Fiber spinning	Maxwell	$\mathrm{De} = 0.01$, $D = 20$	$a(0) = 1$, $v(0) = 1$, $v(1) = D$	Inlet stress $\tau(0)$
Extrusion film casting	PTT	$\mathrm{De} = 0.01$, $\varepsilon = 0.1$, $\psi = 0.1$, $A = 0.5$, $D = 5$	$u(0) = 1$, $L(0) = 1$, $e(0) = 1$, $\tau_{xx}(0) = 1$, $\tau_{yy}(0) = 0.5$, $\tau_{zz}(0) = 0$, $u(1) = D$	Force parameter E

Table 2. Numerical settings used to generate the independent shooting-based reference solutions.

Process	Shooting parameter	IVP solver and settings	Root-finding method	Shooting convergence criterion	Output points
Fiber spinning	$\tau(0)$	MATLAB `ode23tb`; default settings: RelTol $= 10^{-3}$ and AbsTol $= 10^{-6}$	Secant method	$\lvert v(1) - D\rvert < 10^{-3}$	100
Extrusion film casting	E	MATLAB `ode15s`; default settings: RelTol $= 10^{-3}$ and AbsTol $= 10^{-6}$	MATLAB fminbnd	$\lvert u(1) - D\rvert < 10^{-4}$	1000

The output-point count specifies the grid on which the converged reference solution was reported or evaluated. The ODE solver itself uses an adaptive internal integration grid.

Table 3. Architecture and training settings used for the fiber-spinning, baseline EFC, and adaptive-activation EFC PINNs.

Setting	Fiber spinning	Baseline EFC	Adaptive EFC
Network representation	Three separate subnetworks	Single monolithic network	Single monolithic network
Hidden layers and width	4×32 per subnetwork	5×64	5×64
Activation function	tanh	tanh	$\alpha \tanh(\beta x)$, $\beta \in [0.5, 5]$
Initialization	Xavier	Xavier	Xavier
Interior collocation points	301	5000	5000
Collocation-point distribution	Uniform	Logarithmic	Logarithmic
Boundary-condition treatment	Hard output transforms	Soft penalties	Soft penalties
Optimizer	Adam	Adam	Adam
Initial learning rate	10^{-3}	10^{-3}	10^{-3}
Learning-rate schedule	Constant (Adam)	Exponential decay, $\gamma = 0.99$	Cosine annealing
Loss function	$\mathscr{L}_{\mathrm{phys}}$ with $\widetilde{\mathscr{N}_3} = \mathrm{De}\mathscr{N}_3$	$\mathscr{L}_{\mathrm{phys}} + 0.1\mathscr{L}_{\mathrm{bc}} + 10^{-3}\mathscr{L}_{\mathrm{grad}}$	$\mathscr{L}_{\mathrm{phys}} + \mathscr{L}_{\mathrm{bc}} + 10^{-5}\mathscr{L}_{\mathrm{grad}}$; $\lambda_s = 10$
Gradient clipping	None	$\lVert\nabla\rVert_{\max} = 1$	$\lVert\nabla\rVert_{\max} = 2$
Weight decay	None	None	10^{-5}
Maximum training budget	8×10^4 iterations	3×10^4 epochs	30 000 epochs
Early-stopping patience	5000	1000	Fixed training budget
Post-Adam L-BFGS stage	None	Yes	None
Random seeds	5	5	12 attempted; 8 successful

4.2.1 Fibre Spinning PINN Results

We solve the steady fiber-spinning system using a modular PINN architecture in which each field is represented by a separate neural network. The three networks share the same spatial input $x \in (0,1)$ and are trained simultaneously through the coupled residual equations. The network outputs are denoted by

$$\hat{a}(x), \qquad \hat{v}(x), \qquad \hat{\tau}(x),$$

which we collect as

$$\hat{\mathbf{u}}(x) = \begin{bmatrix} \hat{a}(x) \\ \hat{v}(x) \\ \hat{\tau}(x) \end{bmatrix}.$$

The residual components corresponding to Equations (3), (4), and (5) are evaluated by automatic differentiation at interior collocation points $\{x_i\}_{i=1}^{N_f} \subset (0,1)$. To improve optimization conditioning, the constitutive residual is rescaled by the Deborah number,

$$\tilde{\mathcal{N}_3} = \mathrm{De}\,\mathcal{N}_3,$$

which preserves the same root set while reducing imbalance among the residual terms. The motivation is one of magnitude balancing rather than tuning: the steady Maxwell constitutive residual $\mathcal{N}_3 = v\tau' - 2\big(\tau + \frac{1}{\mathrm{De}}\big)v' + \frac{\tau}{\mathrm{De}}$ contains $1/\mathrm{De}$ terms (here $1/\mathrm{De} = 100$) that dominate the $O(1)$ mass and force residuals $\mathcal{N}_1, \mathcal{N}_2$; multiplying through by De gives the algebraically equivalent residual $\tilde{\mathcal{N}_3} = \mathrm{De}\,v\,\tau' - 2(\mathrm{De}\,\tau + 1)v' + \tau$ of $O(1)$ magnitude. This is a rescaling of a single governing equation, not a fitted loss weight, and it is not adjusted against the reference solution. A one-factor sensitivity check confirms it is essential: with the rescaling the model attains total $R^2 = 0.9993$, whereas using the raw residual $\mathcal{N}_3$ with equal residual weights causes training to collapse (mean total $R^2 = -2.14$, with $R_a^2 = -6.9$).

The boundary conditions Equations (6)–(8) are imposed exactly through hard output transforms. Let $\tilde{a}(x)$, $\tilde{v}(x)$, and $\tilde{\tau}(x)$ denote the raw outputs of the three networks. The final approximations are defined by

$$\hat{a}(x) = 1 + x\tilde{a}(x), \qquad \hat{v}(x) = 1 + x(D-1) + x(1-x)\tilde{v}(x), \qquad \hat{\tau}(x) = \tilde{\tau}(x).$$

These transformations enforce $\hat{a}(0) = 1$, $\hat{v}(0) = 1$, and $\hat{v}(1) = D$ exactly, so no boundary-penalty term is required. The training loss is therefore taken as

$$\mathcal{L}_{\text{phys}} = \frac{1}{N_f}\sum_{i=1}^{N_f}\left[\mathcal{N}_1(x_i)^2 + \mathcal{N}_2(x_i)^2 + \tilde{\mathcal{N}_3}(x_i)^2\right].$$

The fiber-spinning PINN uses the modular architecture and training settings summarized in Table 3. All reported results are averaged over five random seeds.

Figure 1 illustrates the modular architecture used for the fiber-spinning system.

Results. Figure 2 compares the PINN predictions with numerical reference solutions. The network reproduces the profiles of $a(x)$, $v(x)$, and $\tau(x)$ accurately over the full domain. Agreement is strongest for the area and velocity fields, while the stress field exhibits slightly larger discrepancies near $x = 1$, where the solution gradient is steepest.

Table 4 reports the R^2 values over five runs. The results are consistently strong across seeds, with mean R^2 values of 0.999762 for $a(x)$, 0.999441 for $v(x)$, and 0.998815 for $\tau(x)$.

Table 4. R^2 values across five runs comparing PINN predictions for $a(x)$, $v(x)$, and $\tau(x)$ against numerical reference solutions.

Run	R_a^2	R_v^2	R_τ^2	**Total** R^2
1	0.999564	0.999089	0.997928	0.998860
2	0.999886	0.999716	0.999462	0.999688
3	0.999819	0.999572	0.999106	0.999499
4	0.999697	0.999220	0.998429	0.999115
5	0.999845	0.999609	0.999147	0.999534
Mean	0.999762	0.999441	0.998815	0.999339
Std Dev	0.000117	0.000242	0.000557	0.000305

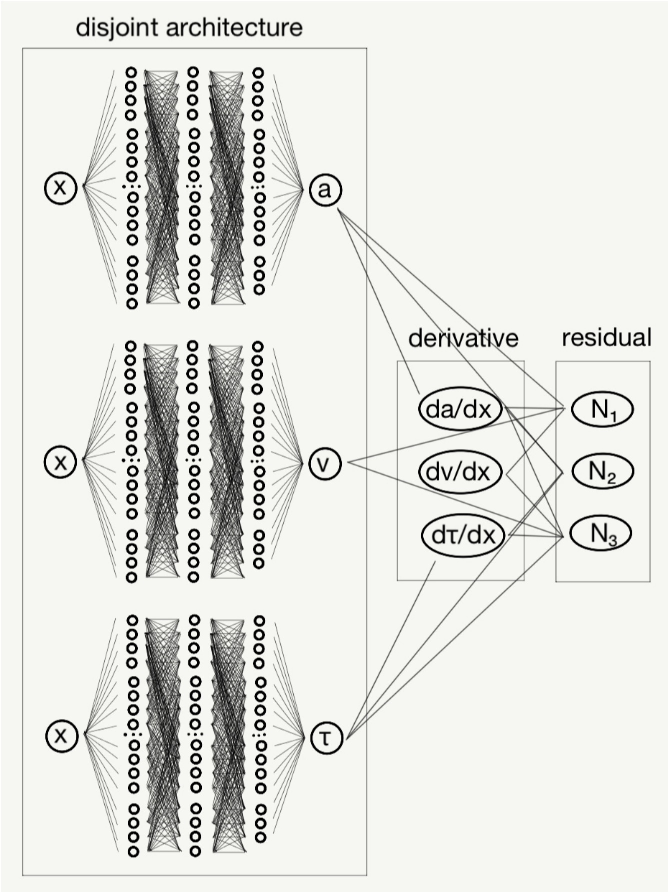

Figure 1. Modular PINN architecture for fiber spinning. Separate subnetworks are used for $a(x)$, $v(x)$, and $\tau(x)$, with coupling enforced through the residual equations during joint training.

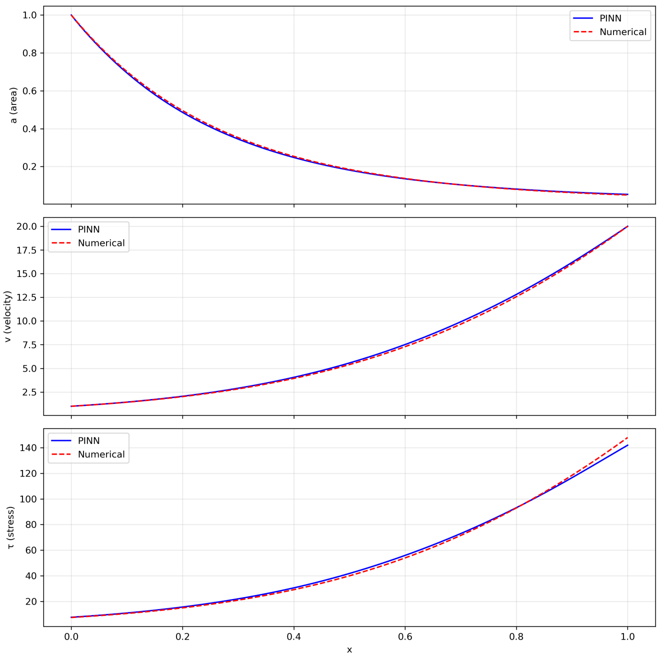

Figure 2. Comparison of PINN-predicted profiles for $a(x)$, $v(x)$, and $\tau(x)$ (mean over five random seeds) against numerical reference solutions on $x \in [0,1]$.

Beyond the coefficient of determination, Table 5 reports the relative L_2 error, maximum absolute error, and boundary MAE for the modular model, using the definitions in Section 2. The mean relative L_2 error is 0.0152 ± 0.0031. Because the inlet area, inlet velocity, and outlet velocity are imposed through hard output transforms, the boundary MAE is exactly zero by construction, in contrast to the soft-constraint EFC models discussed in Section 4. The maximum absolute error (4.85 ± 0.91) is dominated by the stress field, whose magnitude is near the outlet, and is therefore reported per case rather than compared across cases; the relative error is the scale-independent metric used for cross-case comparison.

Table 5. Error summary metrics for the modular fiber-spinning PINN, over five seeds.

Metric	**Mean $\pm$ Std Dev**
Mean relative L_2 error	0.0152 ± 0.0031
Max absolute error	4.85 ± 0.91
Boundary MAE	0 (exact, hard constraints)

Sensitivity to the constitutive rescaling. As noted above, the De-rescaling of the constitutive residual is a magnitude-balancing step rather than a tuned weight. Table 6 makes its role explicit: removing the rescaling and training on the raw residual $\mathscr{N}_3$ with equal residual weights causes the optimizer to fail, whereas the rescaled formulation used throughout this paper attains total $R^2 = 0.9993$.

Table 6. Effect of the De-rescaling of the constitutive residual for fiber spinning (three seeds, full training budget). Without the rescaling, training collapses.

Constitutive residual	R^2_a	R^2_v	R^2_τ	**Total R^2**
De-rescaled ($\tilde{\mathscr{N}_3} = \mathrm{De}\,\mathscr{N}_3$, used here)	0.9997 ± 0.0001	0.9994 ± 0.0003	0.9987 ± 0.0006	$\mathbf{0.9993 \pm 0.0003}$
Raw ($\mathscr{N}_3$, $1/\mathrm{De}$ terms)	-6.907 ± 0.000	0.542 ± 0.000	-0.044 ± 0.000	-2.136 ± 0.000

Modular versus monolithic architecture. A monolithic single-network PINN, in which one trunk with three outputs represents all fields, was also tested for fiber spinning under an identical training budget, boundary treatment, collocation grid, optimizer, early-stopping criterion, and set of five seeds; only the network topology differs. To ensure the comparison is not confounded by parameter count, we consider two monolithic variants: a naive same-size network (a single 4×32 trunk, ≈ 3.3k parameters, roughly one third of the modular model) and a parameter-matched network (a single 4×56 trunk, ≈ 9.9k parameters, matched to the modular model's ≈ 9.8k). The results are given in Table 7. The modular formulation is the most accurate and the most consistent across seeds (total $R^2 = 0.99932 \pm 0.00029$). The naive same-size single network is clearly worse (0.99424 ± 0.00082), but the parameter-matched monolithic network is competitive (0.99860 ± 0.00058), if slightly less accurate and slightly more variable. We therefore conclude that the modular architecture gives the best and most consistent accuracy for this problem, while noting that a parameter-matched monolithic network also performs well; the gap is modest, and it is the naive same-size single network — rather than the monolithic formulation per se — that underperforms.

Table 7. Modular versus monolithic fiber-spinning PINN under an identical training budget and metrics (mean $\pm$ std over five seeds). The parameter-matched monolithic control neutralizes the effect of network size.

Architecture	**Params**	R^2_a	R^2_v	R^2_τ	**Total R^2**
Modular ($3 \times 4{\times}32$, used here)	9,795	0.9997 ± 0.0001	0.9994 ± 0.0002	0.9988 ± 0.0005	$\mathbf{0.99932 \pm 0.00029}$
Monolithic ($1 \times 4{\times}32$)	3,331	0.9974 ± 0.0004	0.9949 ± 0.0007	0.9904 ± 0.0015	0.99424 ± 0.00082
Monolithic ($1 \times 4{\times}56$, param-matched)	9,859	0.9994 ± 0.0002	0.9988 ± 0.0005	0.9976 ± 0.0010	0.99860 ± 0.00058

4.2.2 Extrusion Film Casting

We solve the steady extrusion film casting (EFC) system using a monolithic PINN that jointly approximates all six solution fields,

$$\hat{\mathbf{u}}(x) = \begin{bmatrix} \hat{u}(x) \\ \hat{L}(x) \\ \hat{e}(x) \\ \hat{\tau}_{xx}(x) \\ \hat{\tau}_{yy}(x) \\ \hat{\tau}_{zz}(x) \end{bmatrix}, \qquad x \in (0,1).$$

The unknown parameter E appearing in the stress-evolution equations is treated as a trainable scalar and is optimized jointly with the network parameters.

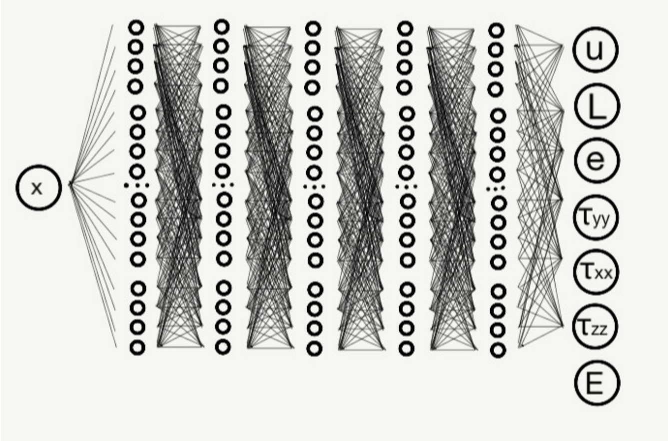

Figure 3. PINN architecture for the EFC system. The network jointly predicts the six coupled fields $u(x)$, $L(x)$, $e(x)$, $\tau_{xx}(x)$, $\tau_{yy}(x)$, and $\tau_{zz}(x)$, while the parameter E is optimized simultaneously.

The baseline EFC PINN uses the monolithic architecture and training settings summarized in Table 3 (see also Figure 3). Interior collocation points are sampled on a fixed logarithmic grid with $N_f = 5000$, and the boundary points are fixed at $x = 0$ and $x = 1$. The inlet and outlet conditions

$$u(0) = 1, \quad L(0) = 1, \quad e(0) = 1, \quad \tau_{xx}(0) = 1, \quad \tau_{yy}(0) = 0.5, \quad \tau_{zz}(0) = 0, \quad u(1) = D$$

are imposed through soft penalty terms. Throughout, the physical parameters are fixed at the values of the numerical reference solution, namely $\mathrm{De} = 0.01$, $\varepsilon = \psi = 0.1$, and draw ratio $D = 5$; the same parameters are used for the baseline and adaptive models so that all reported comparisons are made on an identical problem.

Let $\mathscr{N}_1, \ldots, \mathscr{N}_6$ denote the residuals corresponding to Equations (9)–(14). The physics loss is defined as

$$\mathscr{L}_{\text{phys}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \sum_{\ell=1}^{6} \mathscr{N}_\ell(x_i)^2.$$

The boundary loss is

$$\begin{aligned}\mathscr{L}_{\text{bc}} &= \big(\hat{u}(0) - 1\big)^2 + \big(\hat{L}(0) - 1\big)^2 + \big(\hat{e}(0) - 1\big)^2 \\ &\quad + \big(\hat{\tau}_{xx}(0) - 1\big)^2 + \big(\hat{\tau}_{yy}(0) - 0.5\big)^2 + \big(\hat{\tau}_{zz}(0) - 0\big)^2 + \big(\hat{u}(1) - D\big)^2.\end{aligned}$$

To stabilize training, we also include a gradient penalty

$$\mathscr{L}_{\text{grad}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left[\left(\frac{d\hat{u}}{dx}\right)^2 + \left(\frac{d\hat{L}}{dx}\right)^2 + \left(\frac{d\hat{e}}{dx}\right)^2 + \left(\frac{d\hat{\tau}_{xx}}{dx}\right)^2 + \left(\frac{d\hat{\tau}_{yy}}{dx}\right)^2 + \left(\frac{d\hat{\tau}_{zz}}{dx}\right)^2 \right].$$

The total loss for the baseline model is

$$\mathscr{L}_{\text{total}} = w_{\text{phys}} \mathscr{L}_{\text{phys}} + w_{\text{bc}} \mathscr{L}_{\text{bc}} + w_{\text{grad}} \mathscr{L}_{\text{grad}},$$

with empirically chosen weights

$$w_{\text{phys}} = 1.0, \qquad w_{\text{bc}} = 0.1, \qquad w_{\text{grad}} = 0.001.$$

The baseline optimizer, learning-rate schedule, and early-stopping criterion follow the baseline column of Table 3, including an optional L-BFGS refinement stage.

Baseline results. The baseline PINN is compared against numerical reference solutions in Table 8. The model predicts the velocity and geometric variables accurately, whereas the stress components are less accurate, particularly near the boundaries where the stress gradients are steepest. The inferred value of E converges close to the numerical reference value, although the baseline stress predictions indicate that the monolithic formulation remains harder to train reliably than in the fiber-spinning case.

Adaptive-activation model. To improve stress prediction in the stiff EFC regime, we considered an adaptive activation of the form

$$\phi(x) = \alpha \tanh(\beta x),$$

Table 8. Baseline R^2 values across five runs for the EFC PINN. Total R^2 is the mean over all six variables.

Run	R^2_u	R^2_L	R^2_e	$R^2_{\tau_{xx}}$	$R^2_{\tau_{yy}}$	$R^2_{\tau_{zz}}$	**Total** R^2
1	0.999347	0.977091	0.989178	0.978683	0.931644	0.940303	0.969374
2	0.999353	0.977443	0.989147	0.978651	0.931807	0.940393	0.969466
3	0.999360	0.977324	0.988962	0.978687	0.931855	0.939431	0.969270
4	0.999346	0.977691	0.989183	0.978911	0.931825	0.940492	0.969575
5	0.999347	0.977790	0.989237	0.978513	0.931911	0.940202	0.969500
Mean	0.999351	0.977468	0.989141	0.978689	0.931808	0.940164	0.969437
Std Dev	0.000006	0.000281	0.000105	0.000143	0.000100	0.000374	0.000105

with trainable parameters α and β. In implementation, β is clamped to the interval $[0.5, 5.0]$ during forward passes for numerical stability. The unknown parameter is reparameterized as

$$E = \exp(\log E),$$

which enforces positivity during optimization.

For this adaptive model, the loss is

$$\mathscr{L}_{\text{total}} = \mathscr{L}_{\text{phys}} + \mathscr{L}_{\text{bc}} + 10^{-5}\mathscr{L}_{\text{grad}},$$

where the physics term applies a relative weight λ_s to the three stress residuals. The optimizer, learning-rate schedule, and remaining training settings for the adaptive model are given in the adaptive column of Table 3.

The two loss-weight choices that differ from the baseline — the stress weight λ_s and the reduced gradient-penalty weight 10^{-5} (down from 10^{-3}) — are not tuned against the reference solution, and we assess their influence with one-factor sensitivity studies (each weight varied with all other settings held at the values above; three seeds per setting, full training budget). For the stress weight, Table 9 shows that the results are insensitive to λ_s over a broad range: every value in $\lambda_s \in [1, 20]$ yields total $R^2 \geq 0.993$. Within this plateau the unweighted choice $\lambda_s = 1$ is in fact the most accurate (total $R^2 = 0.9994$), and accuracy decreases mildly as λ_s increases. We therefore do not claim that upweighting the stress residuals improves accuracy; the reported adaptive results (obtained at $\lambda_s = 10$) lie on this plateau and are robust to the choice. For the gradient penalty, Table 10 shows that values $w_{\text{grad}} \leq 10^{-4}$ are statistically indistinguishable from removing the penalty entirely ($w_{\text{grad}} = 0$), whereas the baseline value $w_{\text{grad}} = 10^{-3}$ noticeably degrades stress accuracy (total $R^2 = 0.9795$). This justifies the reduction to 10^{-5} for the adaptive model and shows the mild penalty is otherwise inconsequential.

Table 9. Sensitivity of the adaptive EFC model to the stress-residual weight λ_s (gradient penalty fixed at 10^{-5}; three seeds). Accuracy is robust across $\lambda_s \in [1, 20]$; the reported model uses $\lambda_s = 10$.

λ_s	$R^2_{\tau_{xx}}$	$R^2_{\tau_{yy}}$	$R^2_{\tau_{zz}}$	**Total** R^2	E
1	0.9991 ± 0.0001	0.9995 ± 0.0000	0.9979 ± 0.0001	$\mathbf{0.9994 \pm 0.0000}$	0.1773
5	0.9985 ± 0.0004	0.9994 ± 0.0001	0.9966 ± 0.0010	0.9990 ± 0.0003	0.1774
10	0.9973 ± 0.0009	0.9987 ± 0.0003	0.9921 ± 0.0029	0.9978 ± 0.0007	0.1785
20	0.9920 ± 0.0044	0.9957 ± 0.0022	0.9758 ± 0.0152	0.9931 ± 0.0040	0.1798

Table 10. Sensitivity of the adaptive EFC model to the gradient-penalty weight w_{grad} (stress weight fixed at $\lambda_s = 10$; three seeds). Weights $\leq 10^{-4}$ are indistinguishable from removing the penalty; the old baseline 10^{-3} degrades accuracy. The reported model uses 10^{-5}.

w_{grad}	$R^2_{\tau_{xx}}$	$R^2_{\tau_{yy}}$	$R^2_{\tau_{zz}}$	**Total** R^2	E
0	0.9979 ± 0.0003	0.9988 ± 0.0002	0.9943 ± 0.0007	$\mathbf{0.9984 \pm 0.0002}$	0.1783
10^{-5}	0.9973 ± 0.0009	0.9987 ± 0.0003	0.9921 ± 0.0029	0.9978 ± 0.0007	0.1785
10^{-4}	0.9971 ± 0.0003	0.9979 ± 0.0000	0.9936 ± 0.0006	0.9977 ± 0.0002	0.1779
10^{-3}	0.9804 ± 0.0010	0.9721 ± 0.0002	0.9473 ± 0.0024	0.9795 ± 0.0006	0.1773

Figure 4 and Table 11 show that the adaptive-activation model substantially improves performance across all variables, particularly for the stress fields. Averaged over the successful seeds (defined below), the mean R^2 increases to 0.996867 for τ_{xx}, 0.995979 for τ_{yy}, and 0.991343 for τ_{zz}, while the mean total R^2 rises to 0.997173 (compared with the baseline's 0.969437).

With adaptive activation, the model learns E jointly with the flow variables; across the successful adaptive runs, the inferred value is tightly clustered at $E = 0.178410 \pm 0.000457$. The value of E obtained by the shooting method for this set of parameters was $E = 0.1770$, so the relative error is 0.0080 (that is, 0.80%).

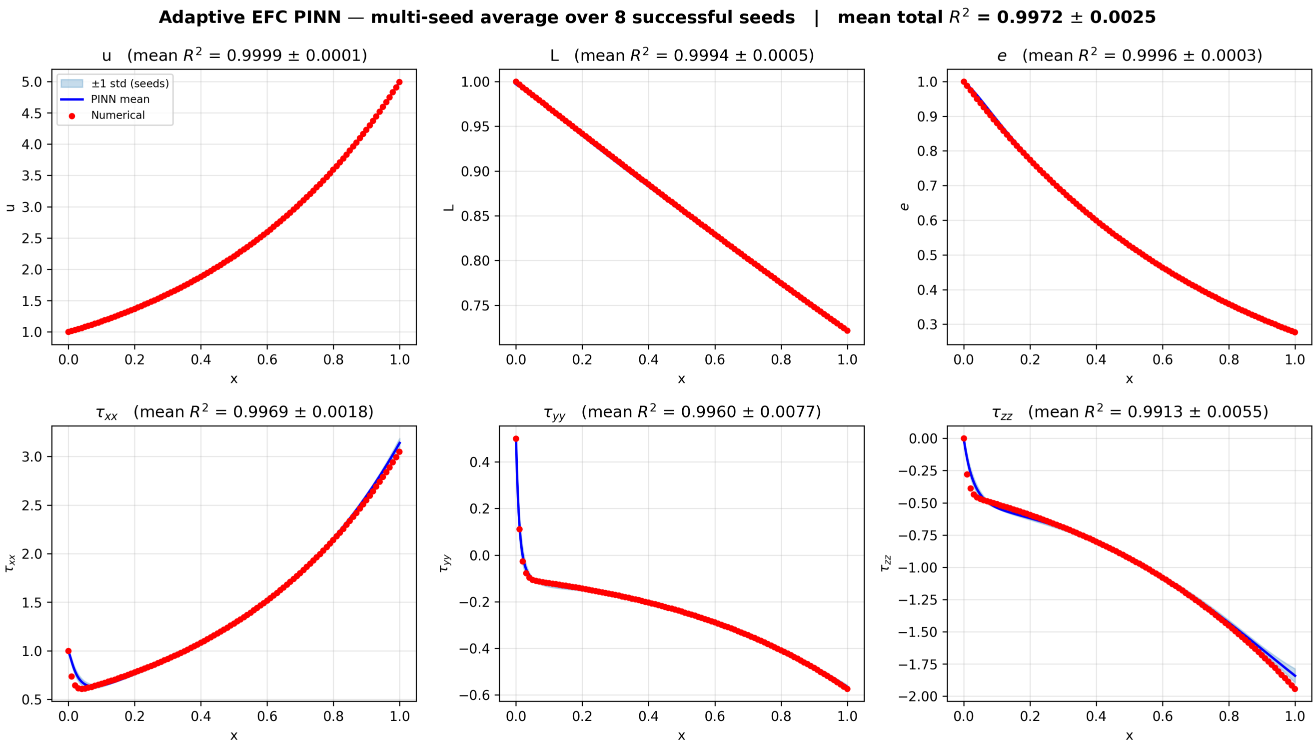

Figure 4. Adaptive-activation EFC PINN predictions on $x \in [0,1]$, averaged over the eight successful seeds. Each panel shows the numerical reference solution, the seed-averaged PINN prediction, and a ± 1 standard-deviation band across seeds. The reported mean total $R^2 = 0.9972 \pm 0.0025$ is the mean over seeds, consistent with Table 11.

Number of runs and success criterion. Because the training of the adaptive model does not converge for every random seed, we state the accounting explicitly. We attempted a fixed, outcome-independent set of twelve seeds and classified each run using a criterion that depends only on the training objective (physics plus boundary residual), not on the reference solution: a run is deemed successful if its converged training loss falls below 10^{-2}. The two outcomes are cleanly separated — successful runs converge to a training loss of $\sim 10^{-3}$, whereas failed runs plateau near 2–3 with unsatisfied boundary conditions — so the threshold is not delicate (a gap of more than three orders of magnitude). Of the twelve seeds, eight (67%) were successful; the four failures (seeds 101, 202, 303, 606) all diverged to a spurious basin with $E \approx 0.27$–0.38 and negative R^2. This is the well-known PINN pathology of convergence to a poor local minimum; since it is detectable from the training loss alone, such a run can simply be discarded and restarted with a new seed. Table 11 reports all eight successful runs, and the failure accounting is summarized in Table 12. We emphasize that no run was excluded using knowledge of the reference solution.

Table 11. R^2 values across the eight successful adaptive-activation runs for the EFC PINN, with inferred E. Runs are the converged seeds from the twelve-seed robustness study (Table 12).

Seed	**E**	R^2_u	R^2_L	R^2_e	$R^2_{\tau_{xx}}$	$R^2_{\tau_{yy}}$	$R^2_{\tau_{zz}}$	**Total** R^2
505	0.177682	0.999992	0.999833	0.999898	0.998667	0.999604	0.997816	0.999302
6789	0.177843	0.999970	0.999933	0.999893	0.999026	0.999611	0.997000	0.999239
1234	0.178263	0.999926	0.999606	0.999701	0.998055	0.998995	0.994381	0.998444
4567	0.178299	0.999946	0.999328	0.999643	0.997800	0.998841	0.994015	0.998262
808	0.178474	0.999904	0.999259	0.999549	0.996411	0.998378	0.991253	0.997459
707	0.178788	0.999802	0.999572	0.999649	0.995787	0.998586	0.988269	0.996944
5678	0.178867	0.999784	0.999274	0.999522	0.996050	0.998227	0.988052	0.996818
404	0.179065	0.999756	0.998325	0.998719	0.993138	0.975586	0.979957	0.990914
Mean	0.178410	0.999885	0.999391	0.999572	0.996867	0.995979	0.991343	0.997173
Std Dev	0.000457	0.000085	0.000466	0.000348	0.001807	0.007723	0.005480	0.002526

Error metrics for both EFC models. Table 13 reports the relative L_2 error, maximum absolute error, and boundary MAE (defined in Section 2) for the baseline and adaptive EFC models, evaluated on the same reference solution and with the same PTT parameters ($\varepsilon = \psi = 0.1$, matching the numerical reference). The adaptive model improves every metric: the mean relative L_2 error falls from 0.058 to 0.016, the maximum absolute error from 0.457 to 0.165, and the boundary MAE from 0.093 to

Table 12. Seed robustness of the adaptive-activation EFC model over twelve seeds. The training-loss success criterion is independent of the reference solution and separates the two outcomes by more than three orders of magnitude.

Quantity	Value
Seeds attempted	12
Successful (train loss $< 10^{-2}$)	8 (67%)
Failed seeds	101, 202, 303, 606
E over successful runs	0.178410 ± 0.000457
Total R^2 over successful runs	0.997173 ± 0.002526
Successful-run training loss	7.3×10^{-4} – 5.3×10^{-3}
Failed-run training loss	2.06 – 3.04 (BCs unsatisfied)

0.0014. The boundary MAE remains nonzero for both models because the EFC boundary conditions are imposed softly, in contrast to the hard-constraint fiber-spinning model whose boundary MAE is exactly zero.

Table 13. Error summary metrics for the baseline and adaptive-activation EFC PINN models (mean $\pm$ std; baseline over five seeds, adaptive over the eight successful seeds).

Model	Mean relative L_2	Max absolute error	Boundary MAE	Total R^2
Baseline ($\varepsilon = \psi = 0.1$)	0.0577 ± 0.0001	0.4570 ± 0.0000	0.0931 ± 0.0001	0.96975 ± 0.00010
Adaptive activation	0.0157 ± 0.0069	0.1647 ± 0.0321	0.00138 ± 0.00092	0.99717 ± 0.00253

4.3 Comparative Discussion

Both the fiber-spinning and extrusion film casting (EFC) systems were solved using PINN formulations tailored to the structure of the underlying models. The two cases illustrate a clear contrast in training behavior. For fiber spinning, the modular architecture with separate subnetworks for $a(x)$, $v(x)$, and $\tau(x)$ produced consistently strong agreement with the numerical reference solution across all variables, with mean total $R^2 = 0.999339$ (Table 4). For EFC, the monolithic network also captured the coupled solution structure, but the baseline model was more sensitive to stiffness in the stress fields: while the velocity and geometric variables were predicted accurately, the stress components were less accurate, giving mean total $R^2 = 0.969437$ (Table 8). With adaptive activation functions and modified training, however, the EFC results improved substantially, reaching mean total $R^2 = 0.997173$ over the successful seeds and yielding strong agreement across both kinematic and stress variables (Table 11). The unknown parameter E also showed good agreement with the numerically computed value ($E = 0.178410 \pm 0.000457$ against the shooting value 0.1770, a relative error of 0.80%). We note that the adaptive training does not converge for every seed — the success rate over a fixed twelve-seed set is 67% — but the failures are identifiable from the training loss alone and are simply restarted; the loss-weight choices that distinguish the adaptive model from the baseline were also verified to lie on a broad accuracy plateau rather than at a finely tuned optimum.

These results suggest that the main difficulty in the EFC problem is not the inability of PINNs to represent the solution, but the greater sensitivity of stress prediction to stiffness, coupling, and steep gradients. In the simpler fiber-spinning system, a modular architecture is sufficient to achieve uniformly strong performance. In the more demanding EFC system, careful design choices—in particular adaptive activations and stress-aware training—are needed to obtain comparable accuracy.

To place these results in context, we also compare the PINN approach with the conventional shooting method used to generate the numerical reference solutions. For both fiber spinning and EFC, shooting is computationally more efficient once a suitable initial guess or parameter range has been identified. In the EFC case, however, the unknown parameter E must be tuned so that the outlet condition is satisfied, and this can be sensitive to the parameter regime. By contrast, the PINN formulation treats E as a trainable variable and infers it jointly with the solution fields during optimization. This avoids the need for a separate shooting search, although at substantially greater computational cost. In the present implementations, the complete shooting calculations typically required on the order of a few seconds once a suitable initial guess or parameter bracket had been specified. PINN training required substantially longer iterative optimization and, for the adaptive EFC configuration, could also require restarting unsuccessful random initializations. Since the computations were not designed as a controlled timing benchmark and wall-clock times were not recorded under identical hardware and software conditions, precise runtime ratios are not reported.

The comparison therefore highlights a trade-off. Conventional numerical methods remain preferable when fast forward solution is the primary objective and a reliable shooting strategy is available. PINNs are more expensive to train, but they offer a unified differentiable framework in which forward solution and parameter inference can be handled simultaneously. In the present study, this is most evident in the EFC problem, where the same model predicts the flow fields and estimates the unknown parameter E within close agreement to the numerical reference value.

Overall, the results indicate that PINNs are effective for these steady viscoelastic boundary-value problems when the architecture is matched to the difficulty of the system. The fiber-spinning problem is handled robustly by a modular formulation, whereas the EFC problem requires more careful training design but can still be solved accurately. A natural direction for future work is to combine the efficiency of classical solvers with the inverse capabilities of PINNs in hybrid numerical–learning frameworks.

5. CONCLUSION

This work investigated Physics-Informed Neural Networks (PINNs) for two canonical steady viscoelastic flow problems in polymer processing: fiber spinning and extrusion film casting (EFC). Both problems are governed by coupled nonlinear boundary-value systems with viscoelastic constitutive structure, and both present difficulties associated with variable coupling, parameter sensitivity, and stiffness in the stress fields.

For the fiber-spinning problem, a modular PINN with separate subnetworks for cross-sectional area, velocity, and stress reproduced the numerical reference solution with consistently high accuracy across all variables. For the more demanding EFC problem, a monolithic PINN was able to predict the coupled velocity, geometry, and stress fields while also inferring the unknown model parameter E. The baseline EFC model captured the kinematic variables well but was less accurate for the stress components, especially in regions with steep gradients. This difficulty was substantially reduced by using adaptive activation functions together with modified training, leading to strong agreement with the numerical reference solution across all variables.

Taken together, these results show that PINNs can provide an effective mesh-free framework for solving structured steady viscoelastic boundary-value problems when the architecture and training strategy are matched to the difficulty of the system. The study also highlights an important distinction between the two case studies: the fiber-spinning problem is handled reliably by a modular formulation, whereas the EFC problem requires more careful design choices because stress prediction is more sensitive to stiffness and coupling. In the EFC setting, the ability to infer the unknown parameter E within the same optimization framework also illustrates the inverse capability of the PINN approach.

At the same time, the results do not suggest that PINNs should replace established numerical solvers in all settings. Classical methods remain preferable when fast forward solution is the main objective and a reliable shooting strategy is available. The strength of the PINN formulation in the present setting is that it provides a single differentiable framework for coupled solution and parameter inference, albeit at higher computational cost.

Future work could extend the present methodology to unsteady viscoelastic flows, broader parameter regimes, and more complex constitutive models. It would also be natural to investigate hybrid numerical–PINN strategies that combine the efficiency of classical solvers with the inverse-modeling flexibility of physics-informed learning.

ACKNOWLEDGEMENT AND FUNDING

The authors receive no financial support for the research, authorship, and publication of this article. The datasets used to evaluate the models in the current study are available at https://github.com/melwina/PINNs-for-viscoelastic-flows, specifically at https://github.com/melwina/PINNs-for-viscoelastic-flows/blob/main/efc/efc_numerical_De0.01_ep0.1_psi0.1_D5.csv. The code used in this study is available at https://github.com/rohanlodhi/pinn_efc_fs.

DECLARATION OF CONFLICTING INTERESTS

The authors declare that they have no competing interests.

REFERENCES

[1] M. Renardy, Mathematical analysis of viscoelastic flows, *Annual Review of Fluid Mechanics*, 21, 1989, 21–34.
[2] T. C. Papanastasiou, C. W. Mocosko, E. Scriven, and Z. Chen, Fiber spinning of viscoelastic liquid, *AIChE Journal*, 33, 1987, 834–842.
[3] M. M. Denn, C. J. Petrie, and P. Avenas, Mechanics of steady spinning of a viscoelastic liquid, *AIChE Journal*, 21, 1975, 791–799.
[4] A. K. Doufas, A. J. McHugh, and C. Miller, Simulation of melt spinning including flow-induced crystallization Part I. model development and predictions, *Journal of Non-Newtonian Fluid Mechanics*, 92, August 2000, 27–66.
[5] T. Barborik and M. Zatloukal, Steady-state modeling of extrusion cast film process, neck-in phenomenon, and related experimental research: A review, *Physics of Fluids*, 32, 2020, 061302.
[6] M. Bechert, Non-Newtonian effects on draw resonance in film casting, *Journal of Non-Newtonian Fluid Mechanics*, 279, 2020, 104262.
[7] D. M. Shin, J. S. Lee, J. M. KIm, H. W. Jung, and J. C. Hyun, Transient and steady-state solutions of 2D viscoelastic nonisothermal simulation model of film casting process via finite element method, *Journal of Rheology*, 51, 2007, 393–407.
[8] B. Debbaut, J. M. Marchal, and M. J. Crochet, Viscoelastic effects in film casting, In *Theoretical, Experimental and Numerical Contributions to the Mechanics of Fluids and Solids*, Boston: Birkhäuser, 1995, 679–698.
[9] N. R. Anturkar and A. Co, Draw resonance in film casting of viscoelastic fluids: a linear stability analysis, *Journal of Non-Newtonian Fluid Mechanics*, 28, 1988, 287–307.
[10] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, 378, 2019, 686–707.

[11] M. Mahmoudabadbozchelou, G. E. Karniadakis, and S. Jamali, nn-PINNs: Non-Newtonian physics-informed neural networks for complex fluid modeling, *Soft Matter*, 18, 2022, 172–185.

[12] S. Thakur, M. Raissi, and A. M. Ardekani, ViscoelasticNet: A physics informed neural network framework for stress discovery and model selection, *Journal of Non-Newtonian Fluid Mechanics*, 330, 2024, 105265.

[13] S. Amini Niaki, E. Haghighat, T. Campbell, A. Poursartip, and R. Vaziri, Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture, *Computer Methods in Applied Mechanics and Engineering*, 384, 2021, 113959.

[14] S. Wang, Y. Teng, and P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, *SIAM Journal on Scientific Computing*, 43, 2021, A3055–A3081.

[15] P. Rathore, W. Lei, Z. Frangella, L. Lu, and M. Udell, Challenges in training PINNs: A loss landscape perspective, *Proceedings of 41st International Conference on Machine Learning*, Vienna, Austria, 2024, 42159–42191.

[16] S. Markidis, The old and the new: Can physics-informed deep-learning replace traditional linear solvers?, *Frontiers in Big Data*, 4, 2021, 669097.

[17] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, *Journal of Computational Physics*, 404, Mar. 2020, 109136.

[18] R. Dhadwal, Numerical study of effect of inertia on stability of fibre spinning, *International Journal of Applied Computational Mathematics*, 2, 2016, 699–711.

[19] R. Dhadwal, S. Banik, P. Doshi, and H. Pol, Effect of viscoelastic relaxation modes on stability of extrusion film casting process modeled using multi-mode Phan-Thien-Tanner constitutive equation, *Applied Mathematical Modelling*, 47, 2017, 487–500.