Informed Machine Learning for Optimizing Melt Spinning Processes

Viny Saajan Victor
Transport Processes Department
Fraunhofer ITWM
Kaiserslautern, Germany
viny.saajan.victor@itwm.fraunhofer.de

Manuel Ettmüller
Transport Processes Department
Fraunhofer ITWM
Kaiserslautern, Germany
manuel.ettmueller@itwm.fraunhofer.de

Andre Schmeißer
Transport Processes Department
Fraunhofer ITWM
Kaiserslautern, Germany
andre.schmeisser@itwm.fraunhofer.de

Heike Leitte
Department of Visualization and Scientific Computing
Rhineland-Palatinate Technical University
Kaiserslautern, Germany
leitte@cs.uni-kl.de

Simone Gramsch
Department of Computer Science and Engineering
Frankfurt University of Applied Sciences
Frankfurt, Germany
simone.gramsch@itwm.fraunhofer.de

***Abstract*—Industrial textiles have increasingly become a major part of our day-to-day lives owing to their various desirable properties. Melt spinning processes are a primary and integral part of the production of these textiles. Optimizing the spinning process while maintaining desirable quality is one of the key challenges for the textile industry. Although numerical models, which are digital twins of physical processes, are often used in optimization, they tend to be computationally expensive for complex scenarios. Hence, in this paper, we utilize machine learning to facilitate the optimization of melt spinning processes. We present a novel, reliable, and informed machine-learning model that is both data- and physics-driven. We further demonstrate the capability of this model to accelerate the optimization and analysis of melt spinning processes.**

***Index Terms*—Industrial textiles, melt spinning, informed machine learning, process optimization, numerical simulation, digital twin, Boundary Value Problems, Ordinary Differential Equations**

I. Introduction

Technical textiles are textile materials crafted and engineered to meet specific performance criteria. Their production aims to instill them with functional attributes like strength, durability, filtration capabilities, and resistance, along with distinctive performance features. These characteristics make them desirable across a multitude of industries, finding application in diverse fields. They are employed in liquid and gas filtration, as seen in items like vacuum cleaner bags and water filtration systems. Furthermore, they play a crucial role in insulation for roofing, flooring, and wall materials, as well as in automotive contexts for seat covers, door panels, and headliners. Medical applications also benefit from their usage, including in surgical gowns, masks, and drapes. Additionally, these fabrics are indispensable components in hygiene products such as diapers, sanitary pads, and wipes. Moreover, they fulfill vital functions in technologies like batteries, fuel cells, and various other domains.

The melt spinning process is the primary and crucial step in the production of technical textiles. In this process, molten polymer is supplied to nozzles positioned at the top of the duct called spinnerets. Viscous filaments are then expelled vertically from the nozzles as shown in Fig. 1. Across a predetermined distance, a flow of cold air is introduced from the side to cool the filaments. The solidified fibers are subsequently gathered by the take-up roller at the bottom. The optimization of the melt spinning process aims at enhancing production efficiency and ensuring the desired quality. This entails various steps, including the optimization of the duct geometry to achieve uniform fiber cooling, which minimizes or ideally eliminates fiber ”dancing” caused by turbulence.

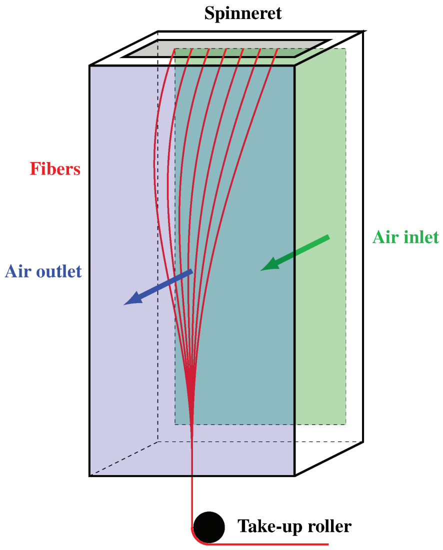

Fig. 1. Sketch of the production of technical fibers using the melt spinning process.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00138

Additionally, optimizing the nozzle positions in the spin pack is crucial to achieve consistent cooling for all fibers.

Controlling the melt spinning process in an industrial environment is often challenging due to its nonlinear nature. The ongoing nature of manufacturing, operating 24/7, coupled with limited types of measurement equipment, imposes constraints on conducting extended experimental series. Consequently, the physical process is typically modeled numerically as digital twins to facilitate systematic analysis within an ideal machine [1]. This analysis entails solving numerical models with varying process, machine, and material parameters. Classical numerical solvers are typically used in simulations. Even though they provide accuracy suitable for engineering purposes, their computational expense makes them unsuitable for real-time applications. Furthermore, these solvers depend on a well-informed initial guess, which is crucial for both solvability and performance. Hence, in this paper, we propose a machine-learning-based model to accelerate the optimization and analysis of the melt spinning process. The main contributions of our work include presenting a novel machine learning model that

- provides a good initial guess for the classical numerical solvers that are used in melt spinning. The goal here is to reduce the convergence time and domain expertise required, resulting in the acceleration of the optimization process.
- provides real-time predictions to accelerate the analysis of the melt spinning process.
- is reliable with the incorporation of process-specific physics knowledge into the training process.

II. Related Work

In recent years, machine learning (ML) has demonstrated remarkable achievements across diverse domains, including computer vision, natural language processing, robotics, and more. This success is attributed to its capacity to process, analyze, and understand complex and varied datasets. The real-time predictability inherent in machine learning makes it well-suited for numerous industrial applications, including the production of technical textiles. Within the textile industry, machine learning models are commonly employed to forecast the quality of textile products based on various parameters that influence the production process. These trained models play a crucial role in the subsequent analysis and optimization of textile manufacturing. Several authors have explored the application of such models for tasks like defect detection and quality estimation [2]–[9]. In alignment with these works, the recent work [10] took a step further by utilizing machine learning to drive a visualization tool that is used to optimize the quality of technical textiles. In the context of melt spinning, the work [11] utilized random forests and neural networks for quality prediction and for identifying the process settings the are leading to abnormal quality data. Additionally, authors in [12] evaluated neural networks with different architectures to predict the magnetic properties from the chemical compositions, melt spinning process parameters, and heat treatment parameters for the production of Sm-Fe-N magnetic powders. Furthermore, [13] used gradient boosting regression to understand the most influential parameters for the coercivity of melt-spun ribbons.

Our approach differs from previous studies in two significant ways. Unlike the machine learning models previously applied in melt spinning, which focused on learning the relationship between process parameters and product quality, our work employs machine learning to address the underlying physics problem. In our methodology, the machine learning model is trained using data generated by a numerical model that represents the physical melt spinning process through differential equations. The goal of the ML model in our approach is to solve these differential equations. The advantage of this approach is that, in addition to facilitating machine learning-based analysis and optimization of the melt spinning process, the predictions generated by the machine learning model can be utilized to enhance the performance of the numerical model upon which it is built.

The second notable difference lies in the fact that previous machine learning models were exclusively driven by data. The drawback of solely relying on data-driven models becomes apparent when there is insufficient data. Such limitations hinder the effectiveness of data-driven machine learning models, especially when the available training data fails to adequately represent variability and capture the system behavior under examination. In instances where the data is noisy and there are no means to impose constraints on the model beyond the data itself, the reliability of the model is compromised. While previous methods in the computer vision domain have focused on adapting training data and the architecture of the ML pipeline to tackle the data scarcity issue [14]–[16], In our work, we focus specifically on the learning algorithm. With the availability of physical laws governing the melt-spinning process formulated as differential equations, we can seamlessly incorporate them into our approach. Furthermore, we assert that the trustworthiness of our model does not solely hinge on the reliability of the training data. By incorporating governing physics into the learning process, our model attains an additional layer of reliability. This approach finds applications across various industrial domains, such as power systems [17], fluid mechanics [18], material defect detection [19], and cardiac activation mapping [20]. A comprehensive review has been conducted that encompasses a wider spectrum of incorporating physics knowledge into machine learning systems, particularly in the form of differential equations [21]. To the best of our knowledge, there has been no baseline work on the application of physics-informed machine learning models for melt spinning processes, making our approach novel for the this domain.

III. Foundations

In collaborations, we frequently work with domain experts in the technical textile industry, including mathematicians and process engineers. Their focus is on modeling and simulating diverse fiber formation processes, such as polymer melt spinning [22] or the manufacturing of glass wool [23]. The underlying fiber models are typically given by systems of differential equations, necessitating the application of numerical solvers. Here, we consider an isothermal viscous uniaxial fiber model (cf. [24]) without aerodynamic forces. In this scenario, the fiber center line is considered as a straight line between fiber inlet r_a and outlet r_b, with a total fiber length of L. The variables of interest in this process are the fiber velocity u and the fiber tension N along the spinline. The system of ordinary differential equations (ODEs) governing the dimensionless velocity u and dimensionless tension N along the spinline is expressed as follows:

$$\frac{du}{dx} = \frac{\mathrm{Re}}{3}\frac{Nu}{\mu}, \tag{1}$$

$$\frac{dN}{dx} = \frac{du}{dx} - \frac{1}{\mathrm{Fr}^2}\frac{\tau_g}{u}, \tag{2}$$

with Reynolds number Re, Froude number Fr and fiber length L:

$$\mathrm{Re} = \frac{\rho_0 u_0 L}{\mu_0}, \tag{3}$$

$$\mathrm{Fr} = \frac{u_0}{\sqrt{gL}}, \tag{4}$$

$$L = \|r_b - r_a\|, \tag{5}$$

for domain $x \in [0, 1]$ with initial condition

$$u(x = 0) = u_{in}, \tag{6}$$

and boundary condition

$$u(x = 1) = u_{out}. \tag{7}$$

In the given equations, μ is the viscosity of the polymer, g is gravity, and τ_g is the fiber direction component parallel to gravity. ρ_0, u_0, and μ_0 denote the reference density, velocity, and viscosity of the polymer, respectively. Furthermore, u_{in} and u_{out} correspond to the dimensionless inlet and outlet velocities.

Although the velocity at the inlet and outlet is known due to the process setup, engineers are particularly concerned with the profiles along the spinline. Excessive fiber tension or overly steep velocity and tension gradients can result in damage to the final product and have adverse effects on fiber properties. In order to optimize the quality of the produced fibers, the velocity and tension profiles need to be analyzed for different parameter settings. This process involves solving the aforementioned system of differential equations for a varied range of parameters. MATLAB's bvp4c solver [25] is commonly employed in solving such boundary value problems. It utilizes a finite difference method combined with a collocation approach, yielding a continuous solution that is fourth-order accurate within the integration interval. This integration interval is divided into smaller intervals through a point mesh. The adaptive subdivision scheme allows for the addition and removal of points as needed. The imposition of boundary conditions and collocation conditions across all sub-intervals results in a global system of algebraic equations. The bvp4c solver is very efficient in solving boundary value problems, providing higher-order accuracy in acceptable computation time for engineering purposes. However, these solvers are not suitable for use in multi-query systems demanding interactive performance (in ms). Additionally, these solvers require a good initial guess of the solution to start with, which affects their solvability and performance. The manual strategy to find an appropriate guess is to divide the original problem into a series of problems by introducing so-called continuation parameters. With the help of these continuation parameters, single terms of the model are set to zero, such that the simplified boundary value problem $bvp_0 := bvp(c_1 = 0, c_2 = 0, ..., c_m = 0)$ is solvable where m is the total number of continuation parameters. Then, the continuation parameters c_i are successively increased. In each continuation step, the solution of the last problem, bvp_{i-1} is used as an initial guess for solving the next boundary value problem, bvp_i, until the original problem, $bvp_n := bvp(c_1 = 1, c_2 = 1, ..., c_m = 1)$ is solvable at final continuation step n. The main challenge here is to find a path through the n-dimensional hypercube of the continuation parameters that leads to a successful simulation of the fiber, which is currently done by domain experts.

IV. Methods and Materials

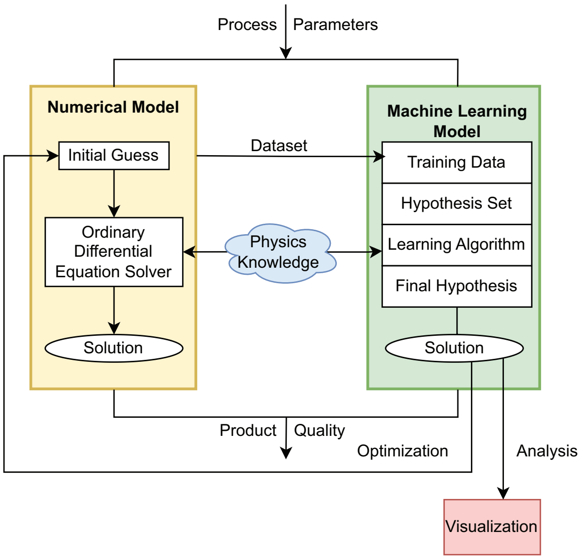

Fig. 2. Workflow of the proposed approach for optimizing the melt spinning process.

In this section, we discuss our proposed approach for integrating a machine learning model into the optimization pipeline of the melt spinning process. Fig. 2 shows the general workflow of the approach. The primary objective of the machine learning model is twofold. Firstly, it aims to provide effective initial guesses for the numerical solver (bvp4c) of the fiber model, thereby minimizing convergence time and reducing the need for domain expertise in providing the initial guess and the continuation parameter strategy explained in the previous section. Consequently, this approach accelerates the optimization process. Secondly, the ML model is designed to provide real-time predictions for varied process parameters, serving as a backend for the analysis tool. As depicted in Fig. 2, we incorporate physics knowledge, represented by differential equations, into the learning algorithm of the ML model. This model, referred to as the informed (data+physics-informed) machine learning model [26], is inspired by the concept of physics-informed neural networks introduced by Raissi et al. [27]. Our contribution reveals that this integration significantly enhances the reliability and performance of ML models. The subsequent subsections elaborate on the construction process of the proposed ML model.

A. Training Data

The input features of the dataset included the parameters of the differential equations (1) and (2). These parameters are the initial and boundary velocities (u_{in} and u_{out}), gravity g, reference density ρ_0, and the x, y, and z components of r_b (with r_a held constant). Additionally, the parameters T, μ_c, B, and T_{VF} of the viscosity function μ, as defined by the below Vogel–Fulcher–Tammann material law, are incorporated.

$$\mu = \mu_c e^{B/(T-T_{VF})} \tag{8}$$

The initial temperature, denoted as T remains constant throughout the process, and the empirical parameters μ_c, B, and T_{VF} are to be determined for the specific fiber material under consideration. Consequently, there are a total of 12 input features. The ranges of the input features were decided in consultation with domain experts as shown in Table I. Latin hypercube sampling [28] was used for generating input data for the data-driven loss. The bvp4c solver was used to solve the differential equations, resulting in output data that consisted of the velocity u and the tension N at grid point x from the solver. Hammersly sampling [29] is used to generate input data for the physics-driven losses. These input data do not require corresponding output data, as the physics-driven losses use unsupervised learning, which will be explained in the later sections.

TABLE I
FEATURE RANGES OF INPUT PARAMETERS

Input Feature	Feature Range
u_{in}	$[0.1, 1.0]$
u_{out}	$[3, 9]$
g	$[5, 15]$
ρ_0	$[800, 2000]$
r_b: x-component	$[0.0, 2.5]$
r_b: y-component	$[0.0, 2.5]$
r_b: z-component	$[0.0, 2.5]$
T	$[500, 600]$
μ_c	$[0.01, 0.5]$
B	$[1500, 2500]$
T_{VF}	$[223.15, 283.15]$
x	$[0.0, 1.0]$

B. Hypothesis Set

We opted for a Deep Neural Network (DNN) as the machine learning model for our approach due to its seamless integration with physics knowledge into the learning algorithm. Additionally, for the physics-driven loss, calculating the first-order derivative of the ML solution at various grid points in the solution domain is a necessity. This process is efficiently and swiftly accomplished through the auto-differentiation functionality integrated in common DNN frameworks. The chosen neural network architecture comprises three hidden layers, each consisting of 150 neurons. The activation function for these layers is the hyperbolic tangent $\tanh$. Initially, training the network involved using the Adam optimizer with a learning rate of 0.001, followed by the L-BFGS optimizer. The impact of various optimizers and learning rates on network convergence is detailed interactively in the 'Training and Convergence' section of the article [1].

C. Learning Algorithm

In this section, we explain how prior physics knowledge in the form of differential equations is incorporated into the learning algorithm. We achieve this by including the physics-driven losses in the DNN loss function. The loss function comprises four loss terms, three of which are physics-driven losses and one is the data-driven loss. The physics-driven losses distinguish themselves from the data-driven loss by not relying on a predetermined ground truth. Instead, they apply penalties to deviations that do not comply with the underlying differential equations. The total network loss L is given by

$$L = L_d + L_i + L_b + L_r. \tag{9}$$

The data loss L_d is calculated as the mean squared error between the ground truth and the predictions for n_d number of data points.

$$L_d := \frac{\lambda_d}{2n_d} \sum_{i=1}^{n_d} \sum_{j=1}^{2} (\mathbf{Y}_{i,j} - \hat{\mathbf{Y}}_{i,j})^2. \tag{10}$$

The initial loss L_i is calculated as the mean squared error between the known initial condition (6) and the predictions for n_i number of initial data points.

$$L_i := \frac{\lambda_i}{n_i} \sum_{i=1}^{n_i} (\mathbf{u}^i_{in} - \hat{\mathbf{u}}^i_{in})^2. \tag{11}$$

The boundary loss L_b is calculated as the mean squared error between the known boundary condition (7) and the predictions for n_b number of boundary data points.

$$L_b := \frac{\lambda_b}{n_b} \sum_{i=1}^{n_b} (\mathbf{u}^i_{out} - \hat{\mathbf{u}}^i_{out})^2. \tag{12}$$

[1] https://observablehq.com/@meltspinning-ws/vis-for-ms

Finally, the residual loss L_r is calculated as the mean squared errors of the residuals from the differential equations (1) and (2) for n_r number of residual data points.

$$L_r := \frac{1}{2n_r}\sum_{i=1}^{n_r} \lambda_u * f_{res}(\hat{\mathbf{Y}}_{i,1}, \hat{\mathbf{Y}}_{i,2}, \hat{\mathbf{X}}_{i,grid})^2 + \\ \lambda_N * g_{res}(\hat{\mathbf{Y}}_{i,1}, \hat{\mathbf{Y}}_{i,2}, \hat{\mathbf{X}}_{i,grid})^2, \tag{13}$$

with residual functions

$$f_{res}(u, N, x) = \frac{du}{dx}(x) - \frac{\text{Re}}{3}\frac{N(x)u(x)}{\mu}, \tag{14}$$

$$g_{res}(u, N, x) = \frac{dN}{dx}(x) - \frac{du}{dx}(x) + \frac{1}{\text{Fr}^2}\frac{\tau_g}{u(x)}. \tag{15}$$

In the above equations, $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ represent ground truth and prediction matrices, respectively. The individual elements $\mathbf{Y}_{i,j}$ and $\hat{\mathbf{Y}}_{i,j}$ correspond to the actual and predicted values, respectively, for the input feature i and the output solution j of the system of differential equations. The matrix $\mathbf{X}$ is the input feature matrix where the individual element $\mathbf{X}_{i,grid}$ corresponds to the grid points of the i^{th} feature vector. The vectors $\mathbf{u}_{in}$ and $\hat{\mathbf{u}}_{in}$ are the actual and prediction vectors of the solution u, respectively, for the initial condition (6). The vectors $\mathbf{u}_{out}$ and $\hat{\mathbf{u}}_{out}$ are the actual and prediction vectors of the solution u, respectively, for the boundary condition (7). The weight coefficients λ_d, λ_i, λ_b, λ_u, and λ_N for data, initial, boundary, residual u, and residual N loss respectively are the model hyper-parameters that need to be tuned during the training phase.

D. Final Hypothesis

In the final stage of our workflow, the predictions from the DNN undergo validation by the classical bvp4c solver in both the optimization and analysis processes. Firstly, in the optimization process, the predictions are used as initial guesses to the bvp4c solver in the final step of the continuation problem $bvp_n := bvp(c_1 = 1, c_2 = 1, ..., c_m = 1)$ discussed in section III. This aims to enhance convergence rates by reducing the time and domain expertise necessary for navigating the continuation parameters. Secondly, in the analysis process, the trained DNN is utilized as a backend for a multi-query visual analysis tool. This tool enables interactive exploration of the parameter space, assisting in navigating and identifying promising regions. It visually guides the user towards local minima. The identified regions of interest can subsequently be subjected to validation using the bvp4c solver.

V. Evaluations

In this section, we analyze the performance of the DNN using various loss functions, as detailed in subsection IV-C. Initially, we trained the network with 500 labeled data points and subsequently with 1000 labeled data points. Table II presents the error metrics for these scenarios across different model types on a test set with 14694 data points. The data-informed network is exclusively trained with the data loss L_d as described in the equation (10) derived from the labeled data points n_d = 500 and 1000. On the other hand, the Physics-informed network is trained solely with the physics losses L_i (using $n_i = 5000$ data points), L_b (using $n_b = 5000$ data points), and L_r with (using $n_r = 50000$ data points) as described in the equations (11), (12), and (13) excluding the labeled data points. In both cases, the hybrid network, which is jointly trained with both data and physics loss, exhibits better accuracy. This highlights the synergy between data-driven and physics-driven models, showcasing their complementary nature in learning to solve parametric differential equations. We trained this network for different values of weight coefficients λ_d, λ_i, λ_b, λ_u, and λ_N from (9). We observed the optimized convergence behavior for $\lambda_d = 1$, $\lambda_i = 1$, $\lambda_b = 1$, $\lambda_u = 0.001$, and $\lambda_N = 0.01$. Figure 3 illustrates the different losses of the hybrid network over 1000 epochs, trained with 1000 labeled data points for the corresponding weight coefficients. Furthermore, the necessity and efficacy of the trained network in the optimization and analysis of the melt spinning process are discussed in the following subsections.

TABLE II
Error Metrics on the Testset for Different Types of Deep Neural Networks

Network Type	**Number of labeled data points**	**MSE**	**MAPE**	**R2-Score**
Data Informed	500	1.0518	0.2966	0.9152
	1000	0.4677	0.1159	0.9610
Physics Informed	-	2.1506	0.2940	0.8204
Data+Physics Informed	500	0.2538	0.0303	0.9806
	1000	**0.1160**	**0.0281**	**0.9910**

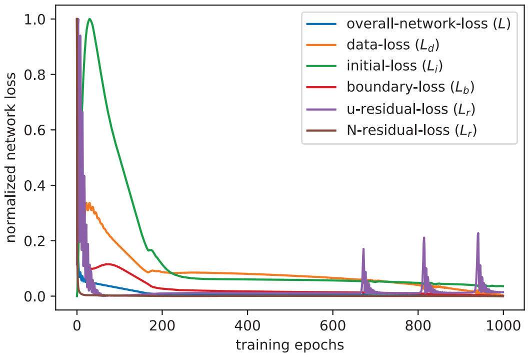

Fig. 3. Convergence of individual losses of a data+physics-informed network over 1000 epochs

A. Acceleration of Optimization

To evaluate the effect of machine learning in accelerating the optimization process of melt spinning, we selected 1948 ordinary differential equations as our test set. The predicted

TABLE III
CONVERGENCE STATISTICS OVER THE ORDINARY DIFFERENTIAL EQUATION TEST SET WITH DIFFERENT INITIAL GUESS STRATEGIES

Initial Guess Strategy	input grid resolution	max grid resolution (NMAX)	Total Differential Equations	Converged Differential Equations	Failed Differential Equations
Constant Guess (constant inlet velocity)	100	1000	1948	1649	299
	1000	1000	1948	164	1784
	1000	10000	1948	1923	25
Linear Guess (between the inlet and outlet velocity)	100	1000	1948	1634	314
	1000	1000	1948	696	1252
	1000	10000	1948	1918	30
Machine Learning (Deep Neural Network)	100	1000	1948	1609	339
	1000	1000	1948	1927	21
	1000	10000	1948	**1929**	**19**

solutions generated by the trained DNN served as an initial guess for the classical bvp4c solver, directly executing the final step of the continuation problem, as detailed in subsection IV-D. For comparative analysis, we included two prevalent classical strategies for initial guess selection: a constant initial solution and a linear initial solution. The convergence statistics for all three strategies are presented in Table III.

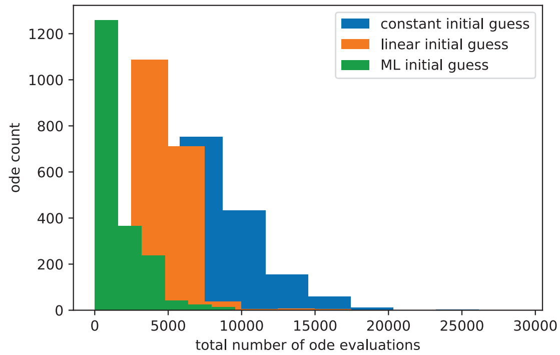

Fig. 4. Total number of ODE evaluations taken by the three initial guess strategies to converge to the final solution.

As indicated in the table, setting the maximum grid resolution in the bvp4c solver (NMAX) to 1000 results in fewer failed cases for the machine learning (DNN) initial guess compared to the constant and linear guesses. Upon examining the reasons for the higher number of failures with the constant and linear approaches, it was discovered that bvp4c can not refine the mesh based on the convergence criteria due to the imposed limitation on NMAX. This limitation also explains why a grid resolution of 100 performs better than 1000 in this context. Notably, the ML initial guess requires less refinement in this case since its values are already closer to the final solution.

Upon increasing NMAX to 10,000, all three strategies exhibit a reduced number of failed cases. However, comparing the total number of ODE evaluations required to converge to the final solution, the ML strategy stands out, demanding fewer ODE evaluations, as illustrated in Fig 4. Furthermore, it becomes apparent that the ML strategy performs consistently well across both lower and higher NMAX values. This observation highlights the advantage gained by employing the ML solution as the initial guess, contributing to faster convergence. As higher NMAX values afford more computational time and space, the effect of the ML strategy becomes particularly evident.

B. Acceleration of Analysis

In this section, we showcase the real-time capabilities of the machine learning (ML) model, demonstrating its suitability for analyzing the melt spinning process. We evaluated execution times on a workstation with a 50-core Intel® Xeon® Gold 6348 (2.60 GHz) CPU. As shown in Table IV the trained ML model demonstrates real-time scalability during the prediction phase, making it suitable for handling extensive datasets. This characteristic proves especially advantageous in the multi-query process, which is essential for the analysis and optimization phases. Consequently, leveraging the trained ML model, we have developed a visualization tool tailored for simulation experts and mathematicians. This tool facilitates a better understanding of the influence of diverse process parameters on crucial attributes related to product quality. Furthermore, it facilitates the identification of local minima along the trajectory of the parameter of interest. The gradient analysis incorporated into the tool proves valuable for identifying significant velocity and tension gradients, providing insights crucial for simulation experts. The primary objective of this tool is to streamline the analysis of the melt spinning process, reducing both the time required and the level of domain expertise necessary for a comprehensive evaluation. The source code, and trained ML model necessary for building and running the visual analysis tool can be accessed with the below-provided link [2].

C. Reliability of the Proposed Workflow

In this section, we discuss the three reliability aspects of the proposed ML model. Firstly, the constructed DNN, trained to solve the differential equations governing the melt

[2] https://github.com/VictorVinySaajan/IsothermalMeltSpinningVisualization

TABLE IV
TIME TAKEN BY THE NUMERICAL MODEL AND THE DNN AT TRAINING AND PREDICTION STAGE FOR ODES

Model Type	Number of data points	Time in seconds
bvp4c solver	500	166.8548
	1000	276.2372
DNN Training	500	2351.6919
	1000	5188.21007
DNN Prediction	500	**0.1812**
	1000	**0.1970**
	10000	0.4427

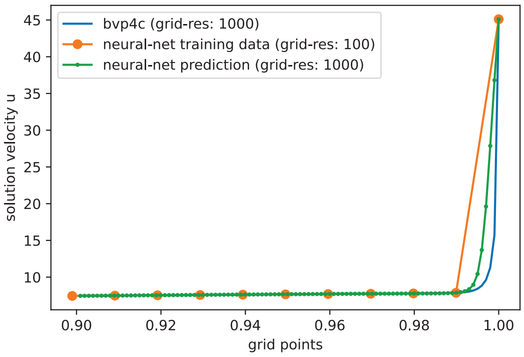

Fig. 5. Comparison of the initial guess made by the deep neural network, trained on a grid resolution of 100, with a prediction resolution of 1000, against the ground truth obtained from bvp4c at a resolution of 1000.

spinning process, is grid-free by construction. This means that a network trained on coarser grids across the solution domain can be used to predict solutions on finer grids. In melt spinning optimization, it is important to predict solutions with finer grid resolutions. This is necessary to capture sudden changes in the solution domain. This is illustrated in Fig 5, where the ML model, trained with 100 grid points, is used to predict solutions on a finer grid of 1000 points. Notably, the network captures the ground-truth solution profile derived from numerical simulations with relatively good accuracy. The figure also explains the reason for the higher failure rate of ML initial guesses at resolution 100 compared to resolution 1000, as highlighted in Table III. This discrepancy is attributed to the inability of the coarser grid to capture abrupt changes effectively. These observations suggest the possibility of exploring grid-refinement techniques similar to those based on the ODE residual employed by the bvp4c solver.

The second aspect of reliability involves integrating physics laws into machine learning models along with data. As demonstrated in Table II, this integration significantly enhances the overall performance of the model. Moreover, we found that this approach improves the model's reliability by not solely depending on the quality of the training data. The hybrid (data+physics-informed) model achieves an additional layer of trustworthiness when faced with suboptimal training data quality. In the ”Reliability Evaluation” section of the interactive article[3], we examined the performance of both the data-informed DNN and the hybrid DNN in the presence of noisy data, outliers, and unseen out-of-the-range data. The results indicate that the hybrid neural network excels in managing noisy data, demonstrates robustness against outliers, and performs better on out-of-the-range data when compared to the exclusively data-driven network.

In the final aspect of reliability, we emphasize that our proposed machine-learning-based workflow undergoes validation with classical numerical solvers in the final phase. Therefore, our approach does not seek to replace the traditional, well-established numerical methods employed in industrial melt spinning optimization. Instead, we utilize machine learning as a catalyst to accelerate the process. As demonstrated in the preceding section, the incorporation of ML reveals substantial benefits in most scenarios, with no associated drawbacks in the worst-case scenario.

VI. CONCLUSION

Our paper introduces a novel machine learning-based workflow designed to accelerate the optimization and analysis of the industrial melt spinning process. Within this framework, we propose a deep neural network driven by both data and the physics laws governing the isothermal melt spinning process. We further elaborated on the construction of this network and its integration into the optimization workflow. We conclusively demonstrated the effectiveness of these networks in accelerating the optimization and analysis processes, emphasizing their reliability aspects. While we demonstrate our workflow primarily with the BVP solver within the framework of isothermal melt spinning processes, this approach has the potential for generalization to other numerical solvers that use an iterative approach starting with an initial guess to solve differential equations.

REFERENCES

[1] R. Wegener, N. Marheineke, and D. Hietel, “Virtual production of filaments and fleeces,” Currents in industrial mathematics: from concepts to research to education, pp. 103–162, 2015.

[2] J. Fun, and L. Hunter, “An artificial neural network model for predicting the properties of worsted fabrics,” Textile Research Journal, vol. 68, no. 10, pp. 763–771, 1998.

[3] R. Beltran, L. Wang, and X. Wang, “Predicting the pilling propensity of fabrics through artificial neural network modeling,” Textile Research Journal, vol. 75, no. 7, pp. 557–561, 2005.

[4] P. H. Yap, X. Wang, L. Wang, and K. Ong, “Prediction of wool knitwear pilling propensity using support vector machines,” Textile Research Journal,SAGE Publications Sage UK: London, England, vol. 80, no. 1, pp. 77–83, 2010.

[5] G. A. Abou-Nassif, “Predicting the tensile and air permeability properties of woven fabrics using artificial neural network and linear regression models,” Journal of Textile Science & Engineering, OMICS Publishing Group, vol. 5, no. 5, pp. 1–6, 2015.

[6] H. E. Eltayib, A. H. M. Ali, and I. A. Ishag, “The prediction of tear strength of plain weave fabric using linear regression models,” International Journal of Advanced Engineering Research and Science, AI Publications, vol. 3, no. 11, p. 236921, 2016.

[3] https://observablehq.com/@meltspinning-ws/vis-for-ms

[7] R. Ribeiro, A. Pilastri, C. Moura, F. Rodrigues, R. Rocha, and P. Cortez, "Predicting the tear strength of woven fabrics via automated machine learning: an application of the CRISP-DM methodology," SCITEPRESS–Science and Technology Publications, 2020.

[8] R. Ribeiro, A. Pilastri, C. Moura, F. Rodrigues, R. Rocha, J. Morgado, and P. Cortez, "Predicting physical properties of woven fabrics via automated machine learning and textile design and finishing features," Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16, pp. 244–255, 2020.

[9] D. Antweiler, M. Harmening, N. Marheineke, A. Schmeißer, R. Wegener, and P. Welke, "Graph-based tensile strength approximation of random nonwoven materials by interpretable regression," Machine Learning with Applications, Elsevier, vol. 8, p. 100288, 2022.

[10] V. S. Victor, A. Schmeißer, H. Leitte, and S. Gramsch, "Visual parameter space analysis for optimizing the quality of industrial nonwovens," IEEE Computer Graphics and Applications, IEEE, vol. 42, no. 2, pp. 56–67, 2022.

[11] A. K. Gope, Y. Liao, and C. J. Kuo, "Quality Prediction and Abnormal Processing Parameter Identification in Polypropylene Fiber Melt Spinning Using Artificial Intelligence Machine Learning and Deep Learning Algorithms," Polymers, MDPI, vol. 14, no. 13, p. 2739, 2022.

[12] H. Hosokawa, E. L. Calvert, and K. Shimojima, "Machine learning prediction for magnetic properties of Sm-Fe-N based alloys produced by melt spinning," Journal of Magnetism and Magnetic Materials, Elsevier, vol. 526, p. 167651, 2021.

[13] X. Tang, L. Li, A. K. Srinithi, H. Sepehri-Amin, T. Ohkubo, and K. Hono, "Role of V on the coercivity of SmFe12-based melt-spun ribbons revealed by machine learning and microstructure characterizations," Scripta Materialia, Elsevier, vol. 200, p. 113925, 2021.

[14] J. Zhao12, J. Li, F. Zhao, S. Yan13, and J. Feng, "Marginalized CNN: Learning deep invariant representations," 2017.

[15] J. Li, S. Xiao, F. Zhao, J. Zhao, J. Li, J. Feng, S. Yan, and T. Sim, "Integrated face analytics networks through cross-dataset hybrid training," Proceedings of the 25th ACM international conference on Multimedia, pp. 1531–1539, 2017.

[16] Y. Xu, Y. Cheng, J. Zhao, Z. Wang, L. Xiong, K. Jayashree, H. Tamura, T. Kagaya, S. Shen, S. Pranata, and others, "High performance large scale face recognition with multi-cognition softmax and feature retrieval," Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1898–1906, 2017.

[17] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," 2020 IEEE power & energy society general meeting (PESGM), pp. 1–5, 2020.

[18] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," Acta Mechanica Sinica, Springer, vol. 37, no. 12, pp. 1727–1738, 2021.

[19] E. Zhang, M. Dao, G. E. Karniadakis, and S. Suresh, "Analyses of internal structures and defects in materials using physics-informed neural networks," Science advances, American Association for the Advancement of Science, vol. 8, no. 7, p. eabk0644, 2022.

[20] F. Sahli Costabal, Y. Yang, P. Perdikaris, D. E. Hurtado, and E. Kuhl, "Physics-informed neural networks for cardiac activation mapping," Frontiers in Physics, Frontiers Media SA, vol. 8, no. 42, p. 42, 2020.

[21] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," Journal of Scientific Computing, Springer, vol. 92, no. 3, p. 88, 2022.

[22] M. Ettmüller, W. Arne, N. Marheineke, and R. Wegener, "On flow-enhanced crystallization in fiber spinning: asymptotically justified boundary conditions for numerics of a stiff viscoelastic two-phase model," Journal of Non-Newtonian Fluid Mechanics, Elsevier, vol. 296, p. 104636, 2021.

[23] W. Arne, N. Marheineke, J. Schnebele, and R. Wegener, "Fluid-fiber-interactions in rotational spinning process of glass wool production," Journal of Mathematics in Industry, SpringerOpen, vol. 1, no. 1, pp. 1–26, 2011.

[24] M. Ettmüller, W. Arne, N. Marheineke, and R. Wegener, "Product integration method for the simulation of radial effects in fiber melt spinning of semi-crystalline polymers," PAMM, Wiley Online Library, vol. 22, no. 1, p. e202200210, 2023.

[25] J. Kierzenka, and L. F. Shampine, "A BVP solver based on residual control and the Maltab PSE," ACM Transactions on Mathematical Software (TOMS), ACM New York, NY, USA, vol. 27, no. 3, pp. 299–316, 2001.

[26] L. Von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, and others, "Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems," IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 1, pp. 614–633, 2021.

[27] M. Raissi, P Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations," arXiv preprint arXiv:1711.10561, 2017.

[28] W. Loh, "On Latin hypercube sampling," The annals of statistics, Institute of Mathematical Statistics, vol. 24, no. 5, pp. 2058–2080, 1996.

[29] T. Wong, W. Luk, and P. Heng, "Sampling with Hammersley and Halton points," Journal of graphics tools, vol. 2, no. 2, pp. 9–24, 1997.