

Meshless PINN-Based Modeling of Hot Forging Processes for Gas Turbine Engine Blade Manufacturing

Viktoriia Fedosieieva^{1,*} and Anton Matiukhin²

¹ Department of Metal Forming, National University Zaporizhzhia Polytechnic, Zaporizhzhia 69063, Ukraine

² Department of Metal Forming, Mechanical Engineering Faculty, National University Zaporizhzhia Polytechnic, Zaporizhzhia 69063, Ukraine

* Correspondence: joxi.victoria@gmail.com; Tel.: +380-97-236-06-60

How To Cite: Fedosieieva, V.; Matiukhin, A. Meshless PINN-Based Modeling of Hot Forging Processes for Gas Turbine Engine Blade Manufacturing. *Journal of Mechanical Engineering and Manufacturing* **2026**. <https://doi.org/10.53941/jmem.2026.100030>

Received: 14 May 2026

Revised: 28 May 2026

Accepted: 3 June 2026

Published: 21 September 2026

Abstract: Blades are critical components of gas turbine engines (GTEs). Precision forging of such components remains computationally demanding because conventional finite element method (FEM) simulations require extensive computational resources and prolonged optimization cycles. The quality of forged blades strongly depends on accurate prediction of the stress-strain state (SSS). Consequently, growing research interest is focused on AI-assisted digital manufacturing approaches. This work establishes the methodological foundations for applying Physics-Informed Neural Networks (PINNs) to precision hot forging processes for GTE blades. The scientific novelty lies in integrating continuum-mechanics equations directly into a Fourier Feature Network-based PINN architecture. The proposed framework enables mesh-free approximation of severe plastic deformation during hot forging. Unlike conventional FEM approaches, the methodology avoids repeated remeshing procedures typical of large-deformation FEM simulations. Automatic differentiation and physics-constrained optimization enable continuous approximation of velocity and strain fields within deforming workpiece domains. The methodology is validated under axisymmetric hot-forging conditions using die-surface collocation points extracted from STL geometry files. One-time network training requires approximately 20 h on standard CPU hardware. After training, the PINN performs field inference over 259,200 spatial points in approximately 1.4 s. Quantitative validation against QForm UK FEM simulations yields an RMSE of 0.557 for the normalized plastic strain profile. The proposed framework is classified at Technology Readiness Level 3 (TRL 3) as a computational proof-of-concept demonstrating the feasibility of PINN modeling for precision hot forging. Compared with conventional FEM simulations, the proposed framework provides an approximately 5000-fold reduction in post-training inference time, excluding the one-time network training cost. These results highlight the potential applicability of the approach to accelerated process optimization, AI-assisted digital manufacturing, and near-real-time digital twin systems for metal forming. No experimental forging data were available for this study. Validation is performed exclusively against QForm UK FEM simulations.

Keywords: stress-strain state; meshless methods; automatic differentiation; metal forming; surrogate modeling

1. Introduction

Blade forging has traditionally been modeled using the FEM. Seminal contributions by Kobayashi [1] and Wagoner [2] established the analytical foundation for large plastic deformation problems. Simulation tools,



Copyright: © 2026 by the authors. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Publisher's Note: Scilight stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

including DEFORM-3D, QForm UK, and ANSYS, accurately predict structural defects and geometric evolution during metal forming. However, high computational complexity remains the primary drawback. When modeling thin blade edges, the mesh often distorts. This necessitates frequent remeshing, which can extend calculation times to hours or even days.

PINNs represent one of the most promising computational directions within the “Industry 4.0” paradigm for precision metal forming. Unlike classical neural networks, PINNs integrate partial differential equations—such as those of equilibrium, energy, and material state—directly into the neural network’s loss function. This approach ensures the physical consistency of the obtained solution.

Unlike FEM, PINN does not require the construction of a classical computational mesh. Instead, it uses a set of collocation points in continuous space. Once trained, the network delivers physically consistent field predictions at inference speeds orders of magnitude faster than FEM, without repeated solver calls. The solution of inverse problems is also possible: PINN allows for determining material parameters (for example, the friction coefficient μ or yield strength σ_y using only data on the press force or shape change).

Therefore, the aim of this study is to investigate the applicability of a PINN for modeling the precision forging process of GTE blades and to evaluate its accuracy relative to FEM solutions for the given problem formulation. The present implementation is restricted to an axisymmetric hot-forging benchmark problem serving as an initial validation stage toward future full-scale blade-forming simulations.

The novelty of this work lies in the application of a physics-informed neural network to the problem of precision forging of GTE blades, which is traditionally solved using FEM-based models, as well as in the systematic quantitative comparison of PINN and FEM in terms of stress–strain fields and key integral process characteristics for this class of forming problems.

Consequently, the present work should be interpreted strictly as a computational proof-of-concept framework intended to evaluate the operational feasibility and physical consistency of PINN formulations prior to industrial deployment or experimental verification. In the absence of experimental forging data, the methodology is validated exclusively against an industry-standard FEM solver. Accordingly, the proposed framework currently corresponds to TRL 3, where an analytical and computational proof of concept has been demonstrated, while physical validation remains the subject of future work.

2. Literature Review

With the advancement of Artificial Intelligence (AI), researchers have begun applying classical neural networks to simulate stamping processes and the field of metallurgy.

Wu et al. [3] present a Deep Neural Network (DNN) architecture for predicting the growth of intermetallic phases (Fe, Al, Si) in the Al-Si coating of 22MnB5 steel during austenitization. The authors used a DNN to map non-linear relationships between heat-treatment parameters and the coating’s chemical composition. The neural network successfully predicts the weight percentage of aluminum, iron, and silicon at every point across the coating cross-section. For training, an energy-dispersive X-ray spectroscopy (EDS) dataset was used, collected at various heating rates and quenching temperatures. The proposed framework demonstrated high accuracy for complex two-stage heating processes, confirming its reliability. However, prediction accuracy decreases slightly due to variations in the as-received coating thickness, which is an external factor not fully accounted for by the network.

Subsequent advancements led to the development of architectures capable of capturing the spatial and temporal specifics of deformation. In the study by Zhao et al. [4], the RUGNN—a surrogate framework based on Graph Neural Networks (GNN) is presented, designed to predict blank deformation during the hot stamping of battery boxes. A key feature is the integration of recurrent message passing with multi-scale graph downsampling and upsampling operations. This architecture allows the model to effectively capture both spatial and temporal patterns of metal deformation under varying tool geometries. The recurrent mechanism enables approximating the stamping process not as a static act but as a dynamic evolution of the material. The authors demonstrate that graph downsampling operations are critical for improving prediction accuracy in graph-based learning tasks. The algorithm is positioned as a promising tool for replacing time-consuming real-time simulations. A critical aspect remains the complexity of constructing graphs for very large objects, which may require significant computational resources during the preparation stage.

Liu et al. [5] propose an innovative framework for sequence planning in sheet metal deformation during the sequential rubber tool forming process. The key element is selecting the optimal tool path via deep learning. The authors demonstrated that simple feature extractors show good results only for single-stage deformation, improving generalization by approximately 15%, but prove ineffective at predicting complex sequential paths. For processes involving two stages (strokes), a cascaded network showed the best performance. However, as

complexity increases to three stages, cascaded networks become less efficient, with performance dropping by over 10%. This prompts the adoption of Long Short-Term Memory (LSTM) models, which can account for both spatial and temporal deformation characteristics. LSTM architecture achieved an impressive accuracy of over 90%, highlighting the critical importance of analyzing the “temporal” sequence of deformation steps rather than static geometry alone. Nevertheless, a weak point remains the local error at specific points, which requires integrating additional local feature extractors. The main limitation of the current approach is the difficulty in generating outputs for shapes that differ significantly from those in the training set. This suggests an insufficient extrapolation capability.

These articles demonstrate the evolution of regression modeling in metallurgy: from simple composition prediction (DNN) to accounting for temporal process memory (LSTM) and complex spatial product structures (GNN). However, such systems possess limited interpretability (“black box” nature). They require thousands of real or simulated examples for training. If data is scarce, the network may produce results that contradict the laws of physics.

The pivotal foundation for modern PINN development was established by Raissi et al. [6]. This study introduced the PINNs framework, which integrates physical laws—expressed as nonlinear partial differential equations—directly into the loss function of a neural network. Unlike purely regressive algorithms (e.g., DNNs and LSTMs), PINNs act as regulators that enforce the fundamental laws of mechanics or thermodynamics. The authors use automatic differentiation to compute derivatives of the input data, allowing the network to “understand” the problem’s physical context during training. The architecture can solve complex equations, such as the Navier-Stokes or Schrödinger equations, and predict the state of a system even with a minimal number of training points. PINNs demonstrate high efficiency in identifying unknown system parameters, such as viscosity and plasticity coefficients, from scattered, noisy experimental data. This work effectively eliminates the barrier of distrust toward AI in engineering by ensuring the interpretability of results and guaranteeing that the model will not produce physically impossible values.

Building on this foundation, Haghighat et al. [7] examined the use of the PINN architecture for solving inverse problems and creating surrogate approaches in solid mechanics. The authors proposed an innovative multi-network configuration that provides a more accurate representation of field variables compared to standard single-network approaches. The method was successfully tested on linear elasticity problems and demonstrated effectiveness in modeling more complex nonlinear processes, particularly von Mises elastoplasticity. Validation was conducted using synthetic data generated via Isogeometric Analysis (IGA) and the FEM. This work demonstrates that PINN is a powerful tool for identifying material parameters by combining network training with the automatic determination of unknown physical constants. Furthermore, Niu et al. [8] presented a mathematical formulation of PINN based on the theoretical foundations of finite-strain elastoplasticity.

Recent journal publications have begun to describe the application of PINN for metal deformation and production management. Ding et al. [9] emphasize that traditional constitutive formulations, whether phenomenological or physical, are increasingly being replaced by intelligent methods based on artificial neural networks, which can accurately predict flow stress with minimal error. The authors note that neural networks effectively address the nonlinearity in the relationships among temperature, strain rate, and flow stress. The study by Dornheim et al. [10] considers constitutive modeling ranging from universal function approximators to advanced platforms. It integrates the physics and complexity associated with transitioning to new frameworks. Munzone et al. [11] propose a two-stage surrogate pipeline for the rapid prediction of stress fields in stamping processes, overcoming the computational intensity of classical finite element methods. The authors compare the effectiveness of standard artificial neural networks (ANNs) and PINNs, integrating equilibrium equations into the latter to improve the physical reliability of the results. Bobbili et al. [12] present a physics-informed neural network structure for modeling energy absorption during high-strain-rate deformation in A286 steel lattices. Sun et al. [13] introduce a new predictive methodology based on a PINN to calculate the temperature field of steel slabs during heating. Direct internal temperature monitoring is not feasible due to the extreme furnace conditions. The integration of thermophysical information directly into the network architecture significantly outperformed traditional numerical methods in computational speed, enabling real-time process simulation.

The current literature on PINN in solid mechanics and metal forming primarily focuses on benchmark problems with relatively simple geometries, such as plane-strain blocks, cylinders, compression specimens, and axisymmetric billets. These serve as an initial validation stage before transitioning to industrial-scale geometries. In particular, Almeida et al. [14] demonstrated the effectiveness of PINNs in accurately predicting stress distributions in a triangular plate. Their study confirmed that PINNs can solve solid mechanics problems without relying on labeled training data, instead using conservation laws and boundary conditions. Recent studies have demonstrated the growing potential of physics-informed machine learning for solving complex constitutive and continuum-mechanics problems. Roy et al. [15] addressed two elasto-plastic problems under plane-strain

conditions: a perfectly plastic plate subject to prescribed displacement and an isotropic, linearly hardening perforated plate under uniaxial tension. Also, Roy et al. [16] demonstrated the applicability of physics-informed neural frameworks for constitutive modeling to solve a non-associative Drucker–Prager elastoplastic solid governed by a linear isotropic hardening rule.

These studies confirm the perspective of integrating governing physical laws directly into neural-network optimization for computational mechanics applications.

Recent advances in digital manufacturing and Industry 4.0 highlight the industrial relevance of process-level optimization and data-driven production strategies. Galib et al. [17] investigate additive manufacturing (AM) technologies for composite materials, highlighting how layer-by-layer fabrication enables flexible and digitally integrated production systems. They conduct a SWOT analysis of major AM methods, including LOM, FDM, SLA, SLS, and DED. The authors report that composite AM can expand the role of 3D printing from prototyping toward manufacturing applications, while also noting that different AM techniques offer distinct advantages depending on application requirements. Pinto et al. [18] analyze the effect of machining parameters and coating strategies on cutting performance and surface integrity, showing how process optimization improves manufacturing efficiency. Guimarães et al. [19] investigated the adoption of Lean and Industry 4.0 tools in industrial environments, demonstrating a strong positive relationship between digital technology integration and productivity gains. Together, these studies indicate that recent manufacturing research—including contributions in the *Journal of Mechanical Engineering and Manufacturing* (JMEM)—tends to focus primarily on process optimization, machining performance, additive manufacturing, and Industry 4.0 adoption.

Recent studies highlight the rapid development of physics-informed neural networks (PINNs) in computational mechanics and their increasing integration with digital engineering frameworks. Ren et al. [20] provide a comprehensive review of PINNs, emphasizing their methodological evolution, theoretical foundations, and interdisciplinary applications in scientific computing and engineering analysis, particularly for solving partial differential equations in data-scarce environments. In a closely related direction, Yang et al. [21] investigate data-driven PINNs from a digital twin perspective and demonstrate their capability to construct mesh-free virtual representations, enable multi-fidelity learning, and support scalable surrogate modeling for parametric flow and engineering systems. These studies suggest that PINNs are increasingly viewed not only as numerical solvers for computational mechanics problems, but also as enabling tools for digital twins and physics-informed surrogate modeling in modern engineering applications.

However, the application of PINN-based frameworks to large-deformation hot forging processes remains limited. A significant research gap exists in developing meshless, data-free physics-informed computational frameworks capable of predicting severe plastic deformation. This work introduces a generalized PINN framework tailored for gas-turbine-engine blade forging. Unlike traditional surrogates, the proposed model integrates continuum mechanics principles directly into the loss architecture. It bypasses the need for labeled data and completely eliminates the mesh entanglement and remeshing bottlenecks typical of conventional FEM approaches in thin airfoil regions.

Therefore, the objectives to be addressed in this article are:

- Development of a PINN architecture capable of handling complex forming geometries, demonstrated on an axisymmetric benchmark as a first validation stage.
- Formulation of a composite loss function. Integration of equilibrium equations, plasticity conditions, and friction boundary conditions directly into the network’s learning algorithm.
- Incorporation of material rheological properties. Tuning the model to account for high-temperature alloys characterized by non-linear stress dependence on strain rate and temperature.
- Implementation of a meshless approach. This method avoids element entanglement when simulating thin sections of the workpiece.
- Ensuring high computational speed.
- Verification of the obtained results. Comparison with results from traditional numerical analysis in QForm UK 7.2.4 software packages.
- Assessment of the framework’s potential as a computational core for Digital Twin systems. It enables near-real-time process monitoring and intelligent control in precision forging applications.

3. Methodology

For the implementation of the framework and conducting the computational experiment, the Python 3.14 programming language (Python Software Foundation, Beaverton, OR, USA) is used along with the following stack of specialized libraries:

- **PyTorch**—an open-source library for machine learning (Meta AI, Menlo Park, CA, USA). PyTorch serves as the primary backend for constructing computational graphs and tensor operations [22].
- **DeepXDE**—a specialized library for working with physics-informed neural networks (Lu et al., USA/University of Pennsylvania). It provides tools for constructing domains, defining Partial Differential Equations (PDE), and generating collocation points [23].
- **Trimesh**—a library for loading and processing 3D meshes (GitHub, San Francisco, CA, USA). Trimesh handles import of complex blade geometries from STL files.
- **NumPy**—a fundamental package for scientific computing (Open-source community, USA/International). NumPy handles all array-level numerical operations.
- **PyVTK**—a tool for exporting results to the VTK format (Herrera, Santiago, Chile).
- **ParaView**—open-source software for visualization (Kitware, Clifton Park, NY, USA). ParaView enables post-processing and visualization of predicted three-dimensional stress and temperature fields.

To train the network, two optimization algorithms are applied:

- **Adam**—an adaptive moment estimation method (Kingma, Ba, San Diego, CA, USA). Adam drives rapid initial convergence toward the loss minimum [24].
- **L-BFGS**—a version of the Broyden-Fletcher-Goldfarb-Shanno method for large networks (Nocedal, Evanston, IL, USA). L-BFGS handles final fine-tuning, exploiting smooth loss-function curvature in PINN formulations [25].

Figure 1 illustrates the conceptual framework of the developed PINN approach, from geometry definition to industrial validation.

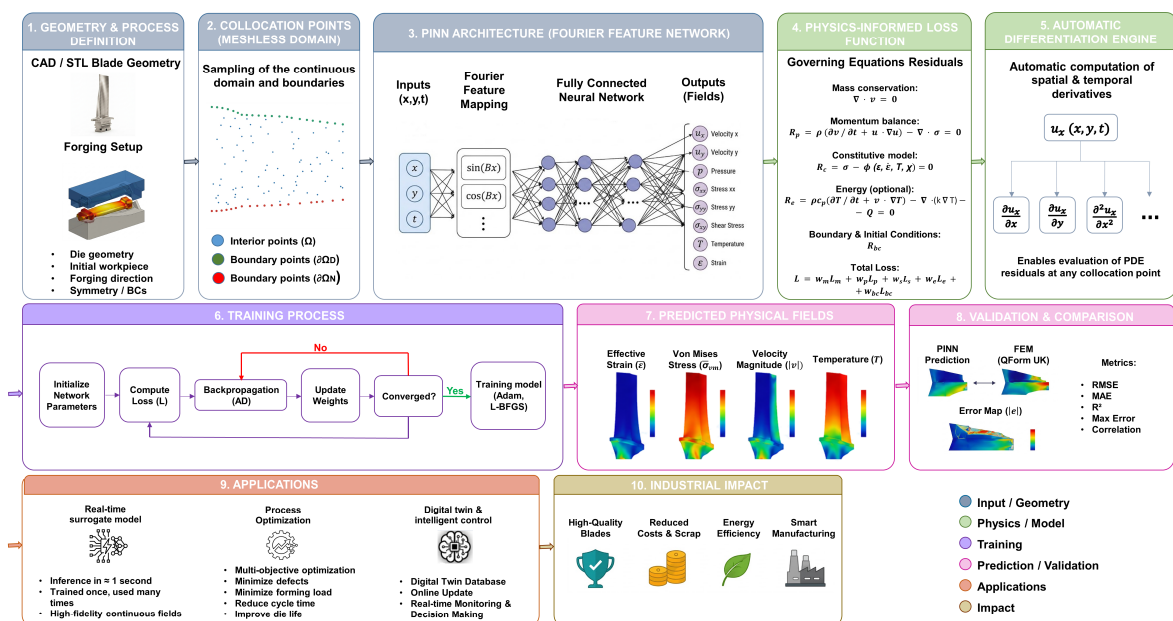


Figure 1. Conceptual framework of the PINN for gas turbine engine blade forging simulation: (1) CAD/STL geometry input; (2) meshless domain collocation point sampling; (3) Fourier Feature Network architecture; (4) composite physics-informed loss function formulation; (5) automatic differentiation engine; (6) multi-phase training workflow; (7) predicted physical fields; (8) validation against FEM; (9) industrial applications; (10) socioeconomic and industrial impact.

The PINN-based forging framework operates through the following sequential stages:

- (1) **Input data.** Coordinates (x, y, z) and deformation time (t) describe the shape of the blade workpiece. For complex shapes, an STL file can be imported. Figure 2 illustrates the longitudinal profile of the blade along the Z-axis. Figure 3 presents the blade design drawing. It includes the cross-section profile (a), longitudinal projection (b), root geometry (c), and airfoil cross-section (d).
- (2) **The neural network approximates the spatio-temporal fields of velocities, stresses, and temperature.** The velocity fields (v_x, v_y, v_z) describe the transient displacement vector of each metal microparticle. This allows tracking of metal flow and die cavity filling. The stress field (σ) determines the internal forces arising in the

workpiece. This is critical for predicting fracture zones. The temperature field (T) accounts for thermal conductivity and the metal's self-heating due to deformation.

- (3) The physical residual quantifies the deviation from the equations of equilibrium or energy.
- (4) Minimization. The network is trained to simultaneously satisfy boundary conditions, constitutive relations, and governing physical laws.

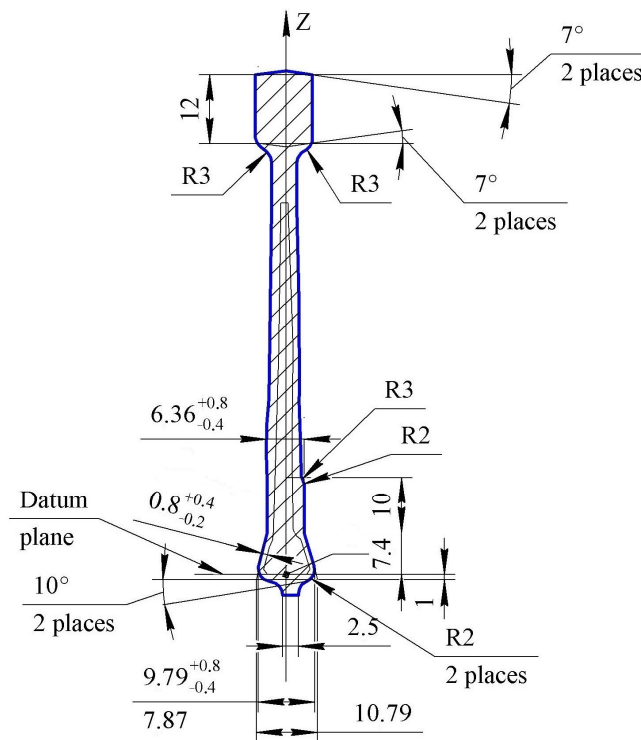


Figure 2. Longitudinal profile of the blade along the Z-axis.

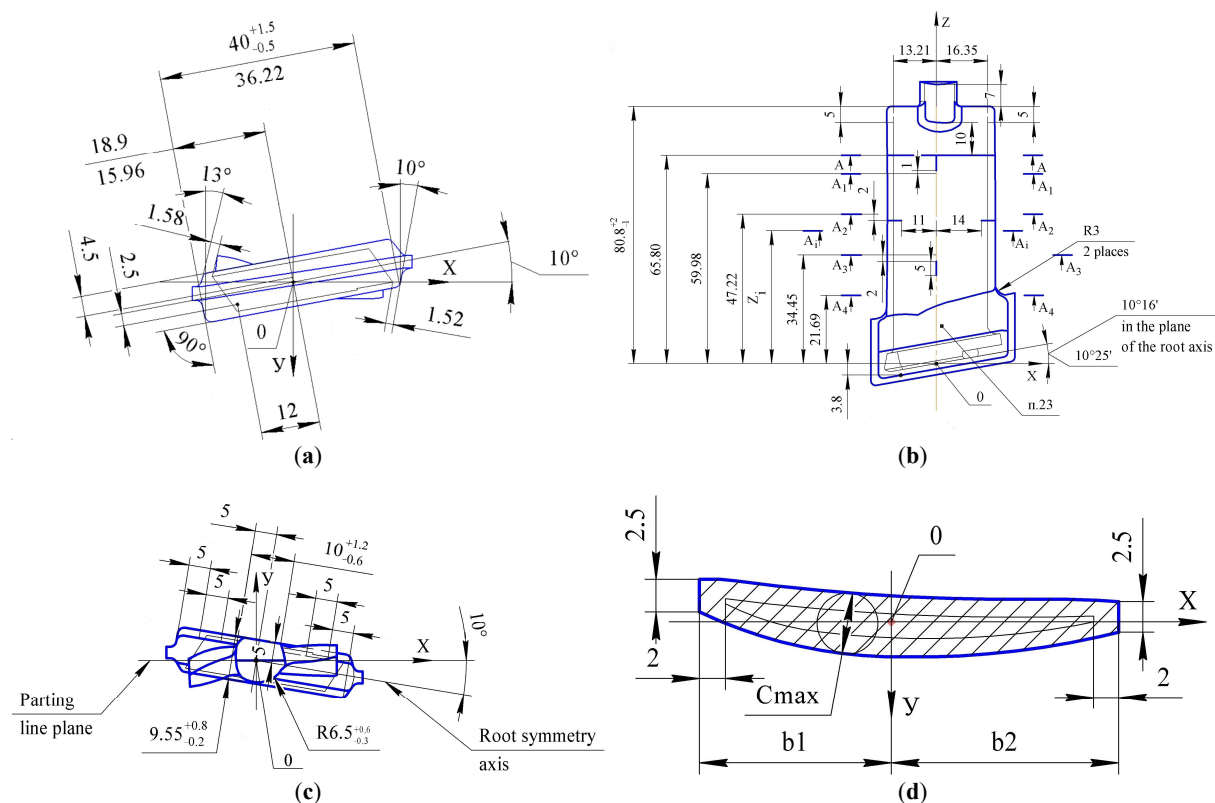


Figure 3. Blade design parameters: (a) profile cross-section in X-Y axes; (b) side view in Y-Z plane; (c) root geometry and section layout; (d) airfoil cross-section.

Unlike finite difference methods, which perform differentiation via spatial discretization, the proposed framework implements a chain rule mechanism within the neural network's computational graph. This allows for obtaining stress tensor components as analytical derivatives via automatic differentiation. Consequently, it ensures mathematical accuracy at any point within the blade's continuous volume.

Then, data collection takes place. The following are required for training:

- boundary conditions: die geometry (from a CAD file) and the press movement speed;
- material properties: the dependence of stress on strain and temperature (rheological curves);
- collocation points: a point cloud inside the workpiece volume and on its surface where the neural network will check the fulfillment of physical laws.

The mathematical formulation of the forging problem comprises an equilibrium equation, the von Mises plasticity condition, frictional contact conditions at the die interface, and, for non-isothermal cases, a heat-conduction equation.

When building the architecture, a MLP is used. The network is described as a functional approximator N , which maps input coordinates to physical quantities (1):

$$y(x, t) = N(x, t; \theta), \quad (1)$$

where $\theta = \{W, b\}$ is the set of network weights and biases.

The input layer (L_{in}) contains 4 neurons (x, y, z, t). For blade forging, the input data must be normalized (scaled to the range $[0, 1]$ or $[-1, 1]$) to avoid gradient vanishing.

The use of 4–8 hidden layers (L_{hidden}) enables the approximation of complex nonlinear dependencies between stresses and strains in regions of stress concentration, particularly at thin blade edges. Beyond network depth, the system's effectiveness depends on the activation function, the distribution of collocation points, and the balance of loss function components.

In the output layer (L_{out}), the number of neurons depends on the completeness of the neural network (typically 7–10):

- metal displacement velocity components: v_x, v_y, v_z ;
- stress tensor components: $\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \tau_{xy}, \tau_{yz}, \tau_{zx}$;
- temperature field: T .

The $\tanh$ activation function is selected for its C^∞ smoothness, ensuring correct higher-order automatic differentiation required for the PDE residual terms.

In order for the network to learn how to correctly model the blade, it is necessary to minimize the weighted sum of the three main components (2), finding network weights that minimize the total error:

$$L(\theta) = w_1 L_{data} + w_2 L_{pde} + w_3 L_{bc}, \quad (2)$$

where L_{data} is the experimental part.

L_{pde} is the physical residual, which includes the equilibrium equations, the incompressibility condition, and the flow law. If the network outputs values where the sum of derivatives in the equilibrium equation does not equal zero, a “penalty” occurs. In the context of PINN, the “penalty” concept mathematically forces the neural network to obey physical laws and form logic. This is achieved by increasing the loss function's value via weight hyperparameters.

L_{bc} represents the boundary conditions. The velocity of the workpiece points at the boundary with the die must equal the movement speed of the die itself. Friction forces on the surface are accounted for using the Siebel or Coulomb schemes. A “penalty” is assessed for metal penetration into the tool.

Weight hyperparameters (w_1, w_2, w_3) determine the priority of each component. The algorithm adjusts them dynamically during training.

The proposed loss function structure allows transforming the boundary-value problem in continuum mechanics into an unconstrained optimization problem. The L_{pde} component acts as a powerful regularizer, ensuring the fulfillment of mass and momentum conservation laws even during intense plastic flow. In such zones, traditional mesh-based methods typically lose solution stability. The weight coefficients w balance the rigidity of physical constraints with compliance with experimental data, providing solution stability even in zones of complex, nonlinear plastic flow regimes.

Training and validation of the architecture take place using the DeepXDE library. For optimization, the Adam algorithm is used for rapid descent, followed by L-BFGS for precise fine-tuning (see Section 4.1 for implementation details). When comparing PINN results with those of a standard simulator (QForm UK, Oxford, UK), PINN should operate significantly faster after training.

The proposed PINN approach allows transforming the mechanics boundary-value problem into an optimization problem (Table 1). Unlike standard neural networks, the PINN enforces physical constraints through its loss function formulation. This constraint enforcement is critical for the aerospace industry, where structural inaccuracies lead to catastrophic component failure.

Table 1. Comparison of PINN and FEM simulation.

Comparison Parameter	Traditional FEM	Proposed PINN
Discretization	Requires mesh construction and adaptation	Meshless (collocation points)
Derivative computation	Numerical (finite difference schemes)	Automatic (exact differentiation)
Inverse problem	Requires multiple iterations	Solved in a single training cycle
Data dependency	Physical constants only	Combination of physics and experiment

The methodology described here is a general conceptual framework for PINN-based three-dimensional forging of GTE blades.

The code presented in this manuscript currently solves a simplified but physically consistent axisymmetric disk forging benchmark, serving as the validation stage prior to full-blade-scale deployment.

4. Results

4.1. Differences between the Implemented Code and the Defined Methodology

The methodology outlined in Section 3 constitutes the general conceptual framework for PINN-based three-dimensional stamping of gas turbine blades. During implementation, several provisions were adapted to address computational constraints and task specifics. Table 2 summarizes the main differences between the defined methodology and the implemented code.

Table 2. Comparison between the defined methodology and the implemented code.

Aspect	Methodology	Implementation
Geometry	3D, coordinates (x, y, z, t) , complex blade geometry	2D, axisymmetry (r, y, t) , cylinder $\text{Ø}175 \times 58 \text{ mm}$
Output Fields	7–10: velocities, stress tensor (6 components), temperature	3: v_r, v_y, ε_p
Rheology	General constitutive material framework	Norton–Hoff
Temperature	Explicit temperature field T as a network output	Isothermal approximation; $T \approx 1150^\circ\text{C}$ via C, m constants
Optimizer	Adam $\rightarrow$ L-BFGS	Adam only, 4 phases: $lr = 10^{-3} \rightarrow 5 \times 10^{-4} \rightarrow 10^{-4} \rightarrow 3 \times 10^{-5}$
Boundary Conditions (BC) on Dies	DirichletBC on domain boundaries	PointSetBC: 1600 points extracted from actual STL files
Network Architecture	MLP, 4–8 layers, $\tanh$	Fourier Feature Network + MLP (7×160), $\tanh$

Problem geometry. A full three-dimensional formulation for a blade requires hundreds of thousands of collocation points and, consequently, weeks of training time. Transitioning to a two-dimensional axisymmetric formulation (r, y, t) reduces computational complexity by a factor of 3–4. Crucially, this simplification still preserves the fundamental physical patterns of the upsetting process. The workpiece is a solid cylinder ($\text{Ø}175 \times 58 \text{ mm}$); the solution in the r – y cross-section uniquely determines the 3D field due to rotational symmetry.

Output fields and stress tensor. The stress tensor is not a direct output of the network; instead, it is calculated within the physics-informed residual function using the Norton-Hoff rheology [26,27]:

$$\sigma_{eq} = C \dot{\varepsilon}_{eq}^m, \quad (3)$$

where σ_{eq} is the equivalent flow stress, $\dot{\varepsilon}_{eq}$ is the equivalent strain rate, and C and m are material constants.

The stress tensor is used exclusively to formulate the equilibrium equations. Direct prediction of the tensor would require additional strain compatibility equations, significantly complicating the task. The chosen approach—a kinematic PINN with integrated rheology—is more robust and sufficient for obtaining the accumulated plastic strain field ε_p .

Temperature field. Explicit modeling of temperature as a separate network output was replaced by an isothermal approximation. The rheological constants $C = 150 \text{ MPa} \cdot c^m$ and $m = 0.12$ correspond to the behavior of carbon steel at approximately 1150°C , which is typical for hot forging. Thus, thermal effect is accounted for indirectly through the material constants. While the current implementation operates under an isothermal approximation, real hot forging is a non-isothermal, thermomechanically coupled process. The proposed PINN framework is modular and inherently scalable to multi-physics problems. Future extensions will incorporate transient heat transfer equations directly into composite loss function residuals.

Optimizer. The L-BFGS method, intended for final refinement, was replaced by a four-phase Adam scheme with a decaying learning rate. This change was caused by a systematic error in the PyTorch line search procedure when calculating the Hessian via the Norton-Hoff function. This instability typically occurred after training stabilized. The four-phase Adam scheme ($10^{-3} \rightarrow 5 \times 10^{-4} \rightarrow 10^{-4} \rightarrow 3 \times 10^{-5}$) ensured convergence: the velocity boundary condition residual bc_vy decreased to 7.9×10^{-5} , confirming the effectiveness of the approach.

Boundary conditions on dies. The standard DirichletBC proved inapplicable because DeepXDE samples boundary points only at the edges of a rectangular domain. However, the contact surface of a shaped die does not coincide with these default boundaries. This was resolved using PointSetBC—an explicit set of 1600 points (80 along $r \times 20$ over t), with coordinates obtained directly from the die STL files. The no-slip condition $v_r = 0$ on the contact surface serves as an analog to the Siebel friction condition (full sticking) and corresponds to typical default settings in QForm UK.

Network architecture. Instead of a standard MLP, a Fourier Feature Network (FourierFNN) was utilized. Input coordinates (x, y, t) are transformed via a random matrix $B \in R^{3 \times 20}$ with $\sigma = 1.0$ into a Fourier encoding $[\sin(2\pi Bx), \cos(2\pi Bx), x]$, which is then fed into an MLP with 7 hidden layers of 160 neurons each and tanh activation. Fourier encoding eliminates the spectral bias effect [28], which is the tendency of standard MLPs to approximate low-frequency components significantly better than high-frequency ones. Overcoming this bias is critical for capturing sharp strain field gradients near the die surfaces. The network scale was selected to provide sufficient capacity to capture high-frequency gradients in severe plastic deformation fields without causing severe gradient vanishing. We selected the hyperbolic tangent (tanh) activation function because it is infinitely differentiable. This property ensures that the second-order derivatives required by the continuous PDE loss terms remain smooth during automatic differentiation.

Alternative architectures commonly employed in PINN frameworks include standard MLPs, sinusoidal representation networks (SIREN), and residual neural networks (ResNets). They also include transformer-based operator-learning approaches. Standard MLP architectures are known to suffer from spectral bias. This bias prevents them from accurately capturing localized high-frequency deformation gradients.

SIREN-based architectures improve high-frequency representation through periodic activations. However, they often exhibit training instability in stiff multi-physics optimization problems involving second-order derivatives. Residual architectures improve gradient propagation in deeper networks. Still, they do not inherently address spectral-frequency limitations.

In the present study, we selected Fourier Feature encoding. This method provides a stable compromise between training robustness, smooth automatic differentiation behavior, and accurate approximation of localized strain gradients near die-contact regions.

4.2. Code Description

During the initial stage, the computational precision is set by default to `float64`. Executing all calculations in double precision is critical for mechanical problems, as the equilibrium equations involve second-order derivatives (Hessians). Using standard `float32` in such cases often leads to unstable gradients or errors. Although this choice doubles the memory requirement and slows down computation by approximately 20%, it is essential for the stability of the PINN solver.

The second stage involves extracting and processing the die geometry from STL files. A dedicated function, `extract_profile`, reads the 3D geometry and extracts a 2D profile in the r - y plane by filtering vertices near the $x = 0$ plane within a specific tolerance. These coordinates are then converted into a polar system where $r = |z|$ and $y = y$. The algorithm identifies the contact boundaries for the upper and lower dies, trims the profile to the workpiece radius ($R_{billet} = 87.5 \text{ mm}$), and generates an array of points representing the physical interface between the tools and the material.

Next, a geometric shift is applied to align the die profiles with the workpiece. Since the original STL files often store the dies in their “press-open” positions (at approximately $y = \pm 197 \text{ mm}$), the coordinates are shifted

so that the contact surfaces meet the top and bottom faces of the workpiece at $y = \pm 29$ mm. This adjustment ensures that the simulation begins with the dies in immediate contact with the initial cylindrical billet.

The fourth stage focuses on scaling and normalizing all physical quantities to dimensionless form. A unified spatial scale based on the billet radius (R_{billet}) is adopted to ensure that the radial and axial scales are consistent. Maintaining a ratio of $s_r = s_y \approx 1.143$ is vital. If these scales were to differ, the components of the strain rate tensor would be imbalanced, causing the incompressibility condition residual to spike and preventing convergence. The resulting problem domain is normalized to $r \in [0.01, 1.0]$, $y \in [-1.0, 1.0]$, and $t \in [0, 1]$, where $t = 1$ represents the completion of the 200 ms forging stroke.

To efficiently determine the y -coordinate of the die surface for any given radial position during boundary condition enforcement, linear interpolators are constructed using `interp1d`. These interpolators allow the framework to retrieve the normalized die height at any point r without repeatedly querying the complex STL geometry, streamlining the calculation of boundary residuals.

The problem's geometry is then formally defined within the DeepXDE framework using a rectangular spatial domain multiplied by a time domain. This `GeometryXTime` object serves as the basis for automatically generating collocation points for Partial Differential Equations (PDEs) and points for boundary and initial conditions.

To strictly enforce known physical constraints, an output transformation layer is applied to the neural network. The radial velocity (v_r) is multiplied by r to automatically satisfy the symmetry condition at the axis ($r = 0$), while the axial velocity (v_y) and plastic strain (ϵ_p) are multiplied by t to ensure they start at zero at the beginning of the process ($t = 0$). Furthermore, a `softplus` activation is applied to ϵ_p to guarantee non-negative values. This hard-coding of constraints eliminates the need for explicit initial conditions for velocities and reduces the associated residuals to machine precision.

The core of the model is the PDE function, which defines the physical residuals the network must minimize. It begins by calculating the strain rate tensor components (e_{rr} , e_{yy} , e_{00} , e_{ry}) using automatic differentiation, with scaling factors applied to convert normalized derivatives back into physical units. These components are then used to enforce the incompressibility condition (mass conservation), which is assigned the highest weight in the loss function because any volume change in a metallic forging process is physically impossible.

Material behavior is governed by the Norton–Hoff rheology, describing hot carbon steel at approximately 1150 °C. The flow stress is calculated based on the strain rate intensity and the rate-sensitivity exponent ($m = 0.12$). This rheological constitutive law determines the effective viscosity, which is then used in the quasi-static equilibrium equations. To compute the stress gradients, second-order derivatives of the velocity fields are required. To prevent memory overflow during the calculation of these Hessians in PyTorch, the differentiation graph for viscosity is detached from the main Hessian graph. This separation is performed via the `detach` method after training stabilizes. Additionally, an evolutionary equation is included to track the accumulation of plastic strain over the physical duration of the process.

Boundary conditions are enforced using `PointSetBC`, which uses a 1600-point grid derived from the die interpolators. The algorithm sets the axial velocity of the dies to ± 1 (normalized) and enforces a no-slip condition ($v_r = 0$), which acts as an analog to the Siebel friction condition for full sticking. To avoid numerical conflicts, the initial time step ($t = 0$) is excluded from these point sets. This exclusion is necessary because the output transformation already forces the velocities to zero at $t = 0$, whereas the actual die movement requires a non-zero initial boundary value.

The network architecture utilizes a Fourier Feature Network (FourierFNN) rather than a standard Multi-Layer Perceptron (MLP). The input coordinates (r , y , t) are transformed via a fixed random matrix into a Fourier encoding, which is then fed into an MLP with 7 hidden layers of 160 neurons each. This approach overcomes the “spectral bias” effect. Consequently, the network successfully captures both the smooth velocity fields within the billet and the sharp strain gradients near the die surfaces, which standard MLPs often struggle to resolve.

The loss function is balanced using a weight vector that prioritizes the most critical physical constraints. The incompressibility condition and the die velocity boundary conditions are assigned the highest weights (10.0 to 50.0), while the equilibrium equations and plastic strain evolution receive lower weights since their natural residuals are more stable. This balancing ensures that the optimizer focuses on maintaining the process's fundamental kinematics before refining the stress-strain distribution.

Training is executed in five distinct phases to ensure stable convergence, a method known as curriculum learning. It begins with a “warm-up” phase that focuses exclusively on the incompressibility condition and boundary movements to establish a valid initial velocity field. Subsequent phases gradually introduce the full PDE residuals and extend the time domain from the first 40% of the stroke to the full duration. By gradually reducing the learning rate across these phases, the model refines the details of the deformation zones, ultimately achieving high accuracy in both the velocity and strain fields.

Throughout the training process, a visualization callback generates VTK files every 500 iterations. This allows the simulation progress to be monitored in real time using ParaView, providing a visual check of the plastic strain and velocity distributions comparable to the post-processing environment of industrial software such as QForm UK.

In the final stage, the 2D axisymmetric solution is expanded into a 3D representation. By rotating the radial solution across 72 angles, a point cloud of approximately 259,200 points is generated to fill the entire cylindrical volume. This data is saved as a structured 3D VTK file, enabling full volumetric analysis of the forged part's stress-strain state.

The presented algorithm should be considered a validation benchmark for applying PINNs to metal-forming problems. Although the current implementation is limited to an axisymmetric configuration, the results demonstrate the feasibility of physics-informed neural networks for hot forging simulations. Furthermore, they provide a solid basis for future extension to full three-dimensional blade geometries.

5. Discussion

Unlike classical machine-learning benchmarks, no standardized public datasets currently exist for PINN-based hot forging simulations involving large plastic deformations and complex contact conditions. Consequently, validation in the present study is performed against established FEM solutions rather than benchmark neural-network datasets.

To validate the proposed approach, we investigate axisymmetric forging of a disk-shaped steel billet (175 mm in diameter and 58 mm in height) at 1150 °C. Figure 4 shows the geometry of the computational domain in normalized coordinates $(r, y, t) \in [0, 1] \times [-1, 1] \times [0, 1]$. The symmetry axis is located at $r = 0$, while the upper and lower dies impose boundary conditions on the normal velocity at $y = 1$ and $y = -1$, respectively. The forming speed is 100 mm/s.

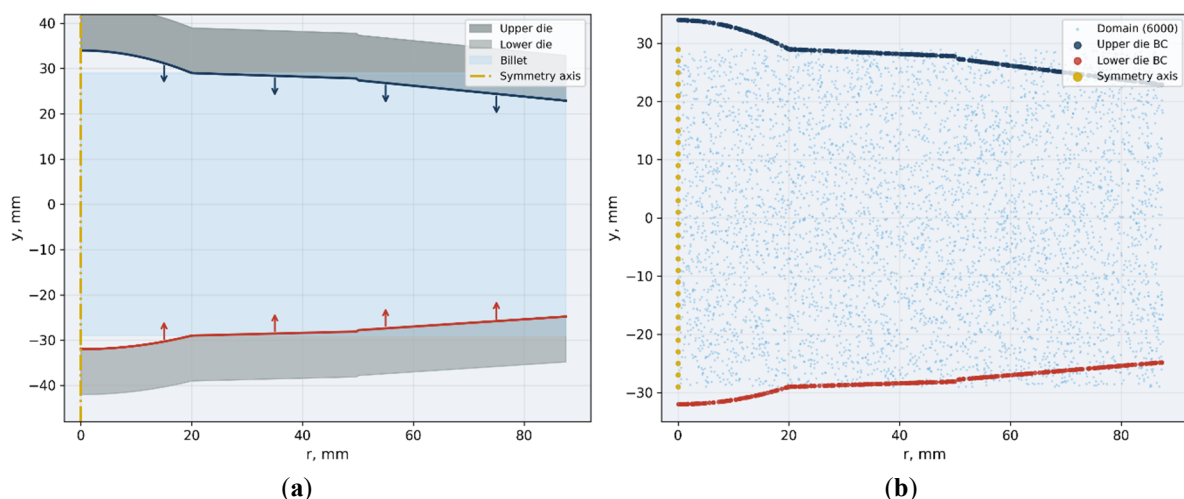


Figure 4. Geometry of the computational domain and collocation point distribution: (a) die profiles and calculation domain; (b) placement of 6000 domain points and 600 boundary condition points. The red and dark blue lines represent the lower and upper die profiles, respectively, while the arrows indicate the normal velocity boundary conditions.

Training employs 6000 domain collocation points and 300 points per die surface (600 total), uniformly distributed to conform to the tool profile. The Fourier Feature Network is trained in five sequential phases following a curriculum learning strategy. The total loss decreased from an initial value of ~ 280 to 0.08, indicating satisfactory convergence. The incompressibility condition is satisfied with an RMS residual of 1.35 s^{-1} , which is acceptable for hot forming problems characterized by large velocity gradients. Figure 5 shows the distribution of accumulated plastic strain ε_p at the end of the process ($t = 1.0$). The computational architecture reasonably reproduces the axisymmetric strain distribution inherent to disk upsetting.

A qualitative comparison with the FEM results from QForm UK (Figure 5c) shows that both approaches yield a radially symmetric strain distribution. This pattern is highly characteristic of symmetric upsetting (Figure 5b). The maximum ε_p value predicted by the PINN is 0.73, whereas QForm UK yields 5.37. This discrepancy arises from differences in the definitions of strain intensity. The PINN provides a dimensionless approximation of accumulated plastic strain. In contrast, QForm UK employs an integral measure calculated along the actual deformation path. To compare the ε_p distribution profiles independently of their absolute scales, both fields were normalized to their respective maximum values. These normalized fields were then evaluated along the equatorial cross-section ($y =$

0, $t = 1.0$). The comparison yields an RMSE of 0.557 (Figure 6a,b). These results indicate a noticeable difference in the radial strain gradient between the PINN and FEM solutions. However, PINN frameworks can successfully reproduce global deformation trends while maintaining consistency with the governing conservation laws, which is appropriate for a proof-of-concept benchmark at this stage of development. The corresponding mean absolute error (MAE) amounts to 0.536, confirming a systematic offset across the entire radial profile.

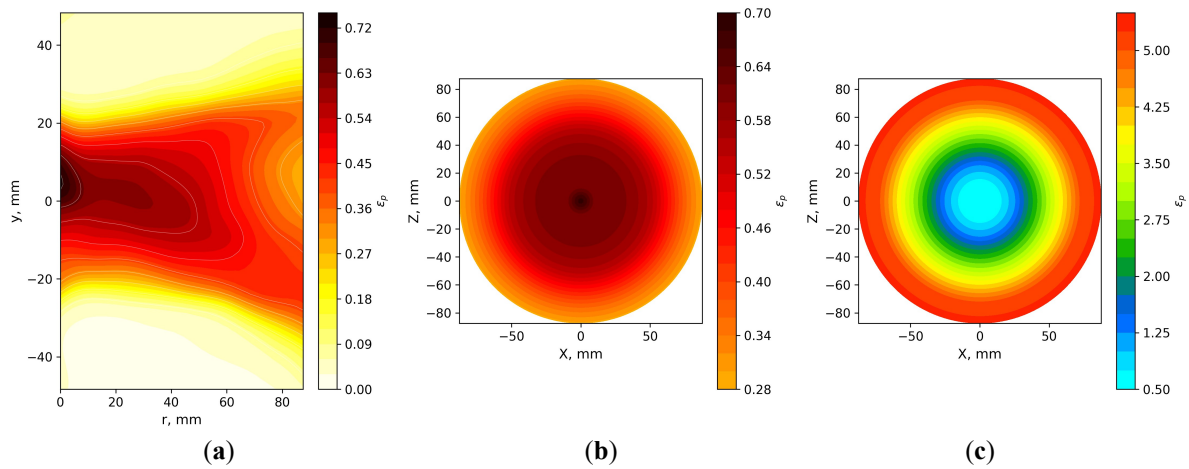


Figure 5. Plastic strain ϵ_p distribution at $t = 1.0$: (a) field in the r - y cross-section; (b) PINN top view; (c) QForm UK (FEM) top view.

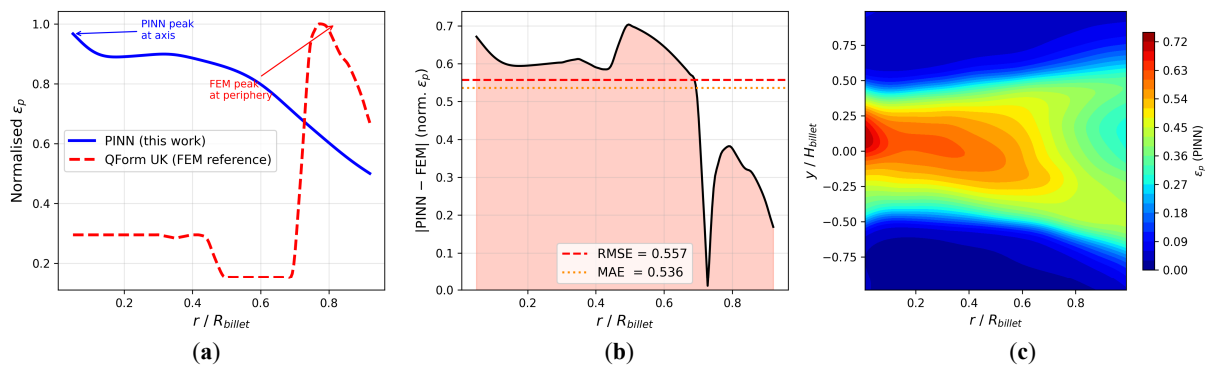


Figure 6. Quantitative validation of the normalized accumulated plastic strain distribution at $t = 1.0$: (a) comparison of normalized radial ϵ_p profiles along the equatorial cross-section ($y = 0$); (b) point-wise absolute error between normalized PINN and FEM profiles; (c) normalized ϵ_p field predicted by the PINN model.

At the current stage, the primary objective of this framework is not to replace high-fidelity FEM certification workflows, but rather to evaluate the feasibility of physics-informed surrogate modeling for accelerated deformation field approximation and future inverse-problem applications.

The PINN framework predicts the maximum accumulated plastic strain near the symmetry axis ($r = 0$), whereas the FEM solution localizes the deformation peak near the die-contact periphery ($r/R \approx 0.8$). This difference results from the simplified velocity-based boundary conditions used in the PINN formulation. Specifically, the current setup does not explicitly account for local die protrusions and geometry-induced contact effects present in the FEM simulation. This spatial inversion of the strain gradient highlights the intrinsic operational differences between discrete and continuous mechanics paradigms. Traditional FEM relies on nodal non-penetration constraints and explicit contact formulation to enforce force balance across tool-workpiece interfaces, naturally trapping high localized shear stresses at the extreme periphery. In contrast, the developed PINN framework simplifies contact mechanics by approximating these boundaries via continuous point-set regularizations (PointSetBC) handled through soft penalty terms in the global loss functional. Because neural networks act as continuous-field approximators governed by low-pass optimization mechanics, they inherently regularize and smooth geometric discontinuities. This continuous soft-boundary regularization rounds off the singular stress gradients at the tool fillets, causing the peak strain localization to migrate toward the internal axis of symmetry ($r \rightarrow 0$) rather than the periphery ($r/R \approx 0.8$).

Importantly, the incompressibility residual of 1.35 s^{-1} and the accurate reproduction of die velocities confirm the mechanical admissibility of the learned solution. Therefore, the observed discrepancy primarily reflects geometric

simplification of the contact conditions rather than failure of the governing equations. Incorporating geometry-aware contact constraints and modified output transforms is identified as the principal direction for future work.

The reported RMSE value of 0.557 for the normalized plastic strain field should be interpreted in the context of early-stage PINN applications in nonlinear continuum mechanics and computational plasticity. In particular, PINN-based approaches for large-deformation and contact-dominated problems are known to exhibit increased approximation error due to strong nonlinearity, strain localization effects, and sensitivity to boundary condition simplifications. At TRL 3, where the framework is evaluated exclusively through numerical benchmarking against finite element solutions without experimental calibration, the obtained error magnitude remains acceptable for an early-stage proof-of-concept benchmark involving strongly nonlinear contact-dominated deformation.

The material flow velocity field and the distribution of the incompressibility residual are shown in Figure 7. The velocity vectors exhibit the physically expected pattern of vertical compression accompanied by radial material outflow from the axis toward the periphery.

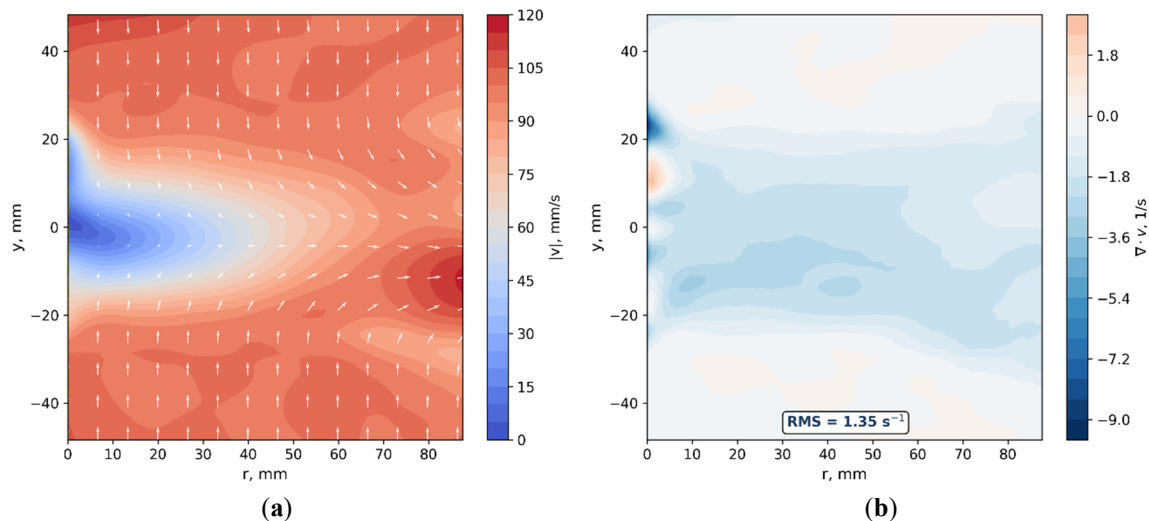


Figure 7. Velocity field and incompressibility: (a) velocity magnitude $|v|$ with flow vectors; (b) $\bar{V} \cdot v$ residual distribution with $\text{RMS} = 1.35 \text{ s}^{-1}$.

The incompressibility condition is satisfied throughout the domain with a root mean square (RMS) residual of 1.35 s^{-1} , indicating progressive enforcement of governing constraints during optimization. Die boundary conditions are accurately reproduced: mean normalized velocity is -1.024 on upper die (target: -1.0) and $+1.003$ on lower die (target: $+1.0$).

PINN inference over 259,200 points requires approximately 1.4 s on CPU. In contrast, a comparable three-dimensional QForm UK FEM simulation may require several hours of computation. This corresponds to an approximate 5000-fold reduction in post-training inference time. The training procedure incurs a one-time computational cost of approximately 20 h (17,000 iterations using `float64` precision with second-order automatic differentiation on an AMD Ryzen 5 3600 6-Core CPU (Advanced Micro Devices, Inc., Santa Clara, CA, USA).

To ensure the computational reproducibility of the established benchmark and provide a quantitative baseline for efficiency analysis, the hardware deployment specifications, volatile memory tracks, and processing time frames are consolidated in Table 3.

Table 3. Quantitative hardware allocation and computational cost benchmark.

Operational Parameter	PINN	QForm UK
Hardware Architecture	AMD Ryzen 5 3600 (6 Cores, 12 Threads @ 3.6 GHz)	AMD Ryzen 5 3600 (6 Cores, 12 Threads @ 3.6 GHz)
Peak Memory Allocation	4.2 GB VRAM/RAM (float64 precision graph)	1.8 GB RAM (Dynamic memory allocation)
One-time Offline Training	20.4 h (17,000 optimization iterations)	N/A (Direct numerical integration setup)
Online Inference/Simulation Time	1.4 s (Across 259,200 domain points)	~6.0 h (Equivalent 2D/3D coupled steps)
Inverse Problem Parameter Calibration	Simultaneous (without additional full FEM recalculation)	High cost (Requires iterative optimization loops)

The reported computational speedup refers exclusively to post-training inference and does not directly replace full FEM preprocessing, solution, and validation workflows.

The convergence analysis indicates stable optimization. The total loss decreased from 280 to 0.08. The velocity residual bc_vy converges from 1.0 to 7.9×10^{-5} , satisfying the threshold below 10^{-4} .

Mesh-density sensitivity analysis shows a 6.6% variation in maximum accumulated plastic strain across 500–24,000 collocation points, indicating near mesh-independence.

Uncertainty quantification and probabilistic confidence estimation remain outside the scope of the present benchmark study and represent important directions for future research.

The Fourier Feature Network-based PINN reproduces physically consistent velocity and plastic strain fields in hot forging. Continuous field representations enable evaluation at arbitrary spatial points with inference times on the order of seconds.

These properties make the framework suitable for Digital Twin integration, enabling near-real-time interaction with process data.

A Digital Twin implementation may include:

- Process monitoring systems (force sensors, die displacement tracking, optical pyrometry) providing real-time boundary and state information.
- Inverse PINN formulations enabling estimation of uncertain parameters such as friction coefficients and non-uniform thermal fields.
- Coupling with predictive control or reinforcement learning strategies for adaptive adjustment of process parameters (e.g., press velocity, counter-pressure) under disturbances such as die chilling or thermal gradients
- Continuous field reconstruction of velocity and accumulated plastic strain, enabling online detection of defects such as underfilling and localized strain concentration in thin geometric regions.

A post-training spatial inference runtime of approximately 1.4 s enables potential integration of the proposed framework into Digital Twin environments and adaptive process-monitoring systems. Real-time process measurements can be incorporated for inverse estimation of friction and thermomechanical parameters, while continuous field reconstruction supports online defect monitoring and process-state evaluation. Compared with conventional FEM workflows, the proposed PINN framework offers significantly faster post-training field approximation for repeated process-state evaluation tasks.

Validation results require critical assessment in the context of industrial forming processes. Transition to industrial use requires addressing limitations in both FEM and PINN formulations.

To visually summarize and contrast FEM and PINN, the complete computational mechanics pipeline of the proposed framework is compared against the traditional finite element method sequence in Figure 8.

FEM remains the benchmarking method for metal forming; however, extreme plastic deformation triggers severe mesh distortion, requiring expensive remeshing and compromising contact accuracy within localized shear zones.

The PINN formulation replaces the mesh with collocation points, avoiding mesh distortion and maintaining solution continuity. Automatic differentiation removes discretization errors associated with numerical gradient evaluation in FEM.

Nevertheless, the inversion of the radial strain gradient (the shift of the peak ε_p towards the axis of symmetry $r \rightarrow 0$ in PINN instead of the periphery $r/R \approx 0.8$ in FEM) indicates that, at the current stage, PINN is inferior to FE models in terms of the accuracy of reproducing stiff nonlinear boundary conditions. FEM strictly guarantees the balance of forces at contact surfaces via variational principles, whereas PINN approximates this balance via soft penalty terms in the composite loss functional.

For a rigorous assessment of scalability, the mathematical and computational limitations of physics-informed learning in nonlinear continuum mechanics must be considered.

Extension of the present 2D axisymmetric formulation to full 3D configurations leads to a substantial increase in computational cost. Adequate resolution of the (x, y, z, t) domain for complex blade geometries requires an increase in collocation points from 6×10^3 to 10^6 – 10^7 . Such an increase leads to excessive GPU memory requirements, partially eroding the computational advantages of PINNs during training.

The optimization of the coupled physics-informed loss function (momentum balance, incompressibility, and frictional contact constraints) is strongly non-convex. Competing constraints induce gradient imbalance, particularly when incompressibility terms are heavily weighted. This suppresses momentum-residual gradients and can lead to convergence to suboptimal minima, with sensitivity to initialization and learning-rate scheduling.

A key computational bottleneck arises from second-order spatial derivatives in the Norton-Hoff constitutive model. Automatic differentiation requires storage of large intermediate tensors, significantly increasing memory

consumption. While graph detachment techniques alleviate memory constraints in the viscosity branch, they introduce approximations that reduce strict consistency of gradient propagation.

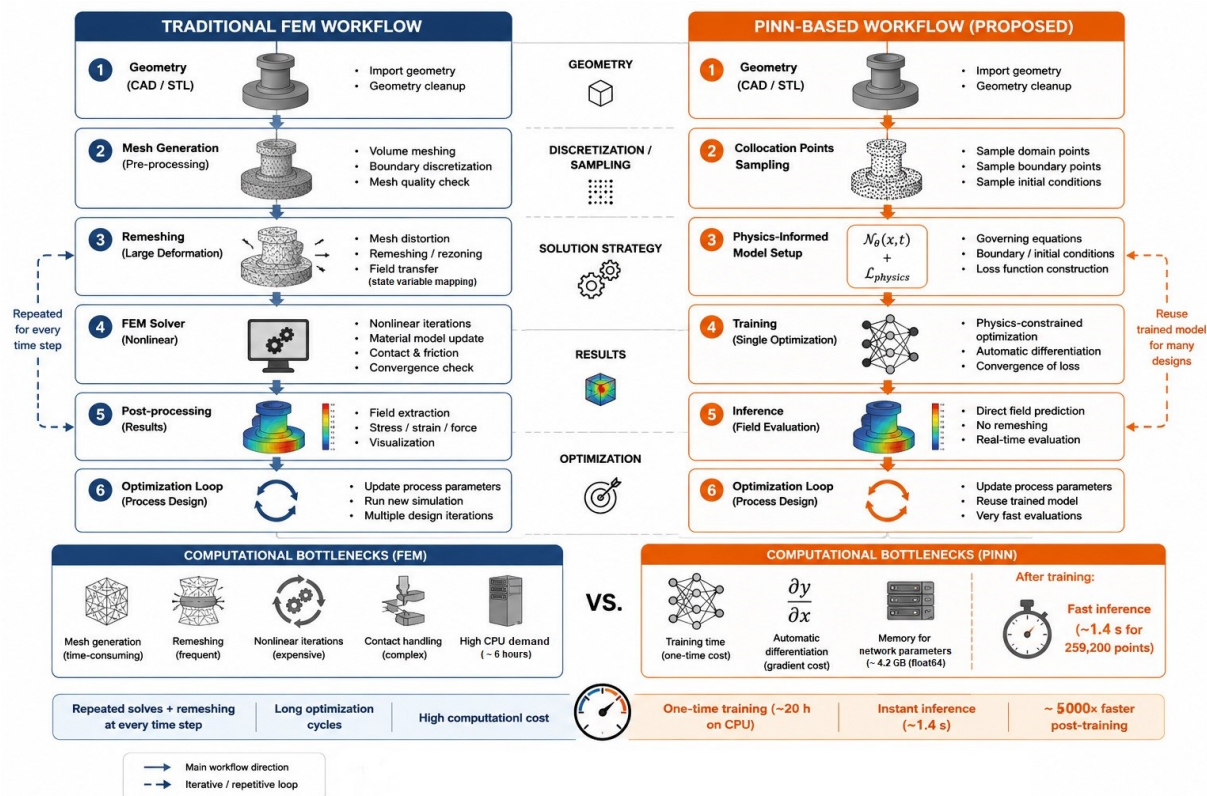


Figure 8. Conceptual workflow comparison between FEM and PINN architecture.

In contrast to conventional FEM, where local accuracy is primarily mesh-controlled, PINNs exhibit a strong non-linear sensitivity to collocation point distribution. Uniform sampling in regions characterized by sharp geometric transitions, such as fillets and narrow tool contact zones, often results in smoothed boundary conditions and reduced local solution accuracy. To mitigate this limitation without uniformly inflating the global collocation point count, future iterations of this framework will implement a Residual-Based Adaptive Refinement (RBAR) strategy. Under this formulation, the network periodically evaluates localized physical PDE residuals across a dense candidate pool after a fixed block of optimization epochs. New collocation points are then adaptively injected strictly into the spatial coordinates exhibiting the highest residual concentrations. This dynamic dense-sampling strategy acts analogously to local mesh refinement (h -refinement) in traditional numerical approaches. By concentrating network capacity directly along severe shear bands, material folding zones, and thin blade-edge regions, RBAR prevents the over-smoothing of high-frequency velocity and strain gradients while fully preserving the global computational efficiency of the data-free training loop.

Classical MLP-based PINNs are also affected by spectral bias, favoring low-frequency components over high-frequency localized gradients. In hot forging, velocity gradients near frictional interfaces exhibit near-singular behavior. Fourier Feature Networks partially mitigate this limitation; however, residual high-frequency errors may still produce non-physical stress oscillations under extreme deformation regimes.

A trained PINN functions primarily as a physics-informed interpolator within a fixed geometric and thermomechanical domain. Changes in geometry or boundary conditions significantly degrade predictive performance, often requiring full retraining, with training times ranging from several hours to more than ten hours. This limits applicability in rapidly changing design or process optimization scenarios.

Finally, extension of the PointSetBC framework to non-axisymmetric GTE blade geometries introduces additional complexity due to spatially varying contact normals and fully three-dimensional material flow in blade-lock regions, where simplified interpolation schemes become insufficient for accurate contact representation.

To systematically position the developed kinematic PINN framework within the broader landscape of modern computational mechanics, its capabilities must be evaluated against both traditional numerical techniques and alternative machine learning paradigms. Table 4 provides a comprehensive state-of-the-art comparison of these

methodologies based on key operational criteria, including boundary fidelity, physical consistency, and computational efficiency.

Table 4. Comparative overview of numerical and machine-learning approaches for metal forming simulations.

Criterion	Finite Element Method (FEM)	Artificial Neural Network (ANN)	Graph Neural Network (GNN)	Long Short-Term Memory (LSTM)	Physics-Informed Neural Network (PINN)
Mesh Dependency	High (Requires strict element connectivity & remeshing)	None (Data point regression)	Moderate (Requires graph-based spatial discretization)	None (Operates on time-series sequences)	None (Uses collocation-point discretization without mesh connectivity)
Physical Consistency	High (Strictly enforced through variational numerical formulation)	Limited (Dependent on training data and constraint formulation)	Low (Dependent on training data boundaries)	Limited (Primarily sequence-oriented without embedded physical constraints)	High (Enforced through continuous differential PDE loss terms)
Computational Cost (Training/Setup)	Moderate (Requires geometry preparation, meshing, and contact definition)	High (Requires massive synthetic dataset generation)	Very High (Complex multi-scale graph building)	High (Requires long sequential histories)	Moderate to High (One-time data-free physics optimization loop)
Computational Cost (Inference)	High (Hours to days for complex 3D topologies)	Extremely Low (Sub-second matrix multiplication)	Low (Fast graph message-passing steps)	Extremely Low (Fast recurrent evaluation)	Low (Fast continuous-field inference after training)
Training Data Requirement	Zero (Requires only physical constants)	High (Thousands of pre-computed FEM examples)	High (Diverse tool geometry profiles)	High (Sequential temporal deformation paths)	Does not require labeled supervised datasets (physics-constrained optimization)
Inverse Problem Capability	Low (Requires expensive multi-iteration optimization loops)	Moderate (Requires specialized inverse architectures)	Low (Constrained by graph topology changes)	Low (Incapable of out-of-sequence backtracking)	High (Supports integrated inverse-parameter optimization)
Industrial Scalability	High (Industry standard, limited to offline validation)	Limited generalization outside the training domain	Moderate (Scales well across geometric families)	Moderate (Limited to simple recurrent steps)	High (Potential applicability for Digital Twin systems)

Despite the identified limitations, several industrial application scenarios remain promising for the proposed PINN framework. In early process design and die development stages, PINNs can act as high-speed surrogate filters, rapidly excluding non-viable process configurations prior to detailed FEM analysis.

A particularly relevant application is the inverse identification of poorly accessible material and process parameters, including friction coefficients and rheological constants. Integration with force sensing and process monitoring systems enables near-real-time calibration of constitutive models by minimizing discrepancies between measured and predicted deformation responses.

The framework is also well suited for Digital Twin and intelligent manufacturing environments. Owing to its low inference cost after training, the PINN model can serve as a computational core for predictive press-control systems, enabling adaptive adjustment of ram velocity and process trajectories to mitigate defects such as overheating, underfilling, or excessive strain localization.

Table 5 summarizes the strategic distribution of application domains between conventional FEM approaches and emerging physics-informed AI methodologies in forging production environments.

A highly promising future direction for the proposed physics-informed neural network framework lies in the direct integration of metallurgical and microstructural evolution kinetics. While the current computational formulation successfully maps macroscopic stress-strain fields under isothermal operations, high-fidelity digital twins of GTE blade manufacturing must simultaneously capture microstructural alterations. The modular architecture of the developed PINN enables seamless inclusion of additional differential equations governing grain growth, dynamic recrystallization (DRX), creep behavior, and phase transformations. Furthermore, incorporating damage evolution criteria (such as void nucleation and growth metrics) directly into the physics-guided loss functional will enable the neural network to predict localized material degradation. Developing a unified PINN

framework to concurrently approximate severe plastic deformation, transient thermal profiles, and real-time microstructural variables is critical. Such a framework represents a promising direction for overcoming some limitations of classical phenomenological approaches.

Table 5. Strategic division of application scopes between FEM and PINN in forging production.

Production Scenario/Task	Preferred Method	Technological Rationale
Final verification and certification of manufacturing processes	FEM	Guarantees absolute accuracy in localizing peak stresses and ensures strict force balance
Multi-objective optimization of die geometry	Hybrid (PINN→FEM)	PINN rapidly narrows down the search space for optimal solutions, while FEM performs targeted final validation
Identification of friction and rheological parameters (Inverse problems)	PINN	Determination of unknown constants within a single training cycle without iterative FE-recomputations
“Digital Twin” monitoring systems (Real-time)	PINN	Near-real-time inference for online monitoring and control of equipment parameters.

6. Conclusions

- We developed and evaluated a Fourier Feature-based Physics-Informed Neural Network (PINN) framework as a proof-of-concept benchmark for axisymmetric hot forging toward future gas turbine engine blade applications. The proposed mesh-free formulation integrates equilibrium equations, incompressibility constraints, constitutive rheology, and friction-related boundary conditions directly into the composite loss function.
- The implemented framework successfully reproduced physically consistent velocity and accumulated plastic strain fields for an axisymmetric upsetting problem under hot-forming conditions. Validation against QForm UK FEM simulations yielded an RMSE of 0.557 for the normalized plastic strain distribution. Although discrepancies remain in localized strain-gradient prediction near die-contact regions, the PINN solution successfully preserved the global deformation behavior and governing physical constraints.
- The proposed approach eliminates mesh entanglement issues typical of large-deformation FEM simulations and enables continuous field approximation over arbitrary spatial locations. After completion of offline training, the framework achieved inference over 259,200 spatial points in approximately 1.4 s on standard CPU hardware.
- The obtained results demonstrate the feasibility of applying physics-informed neural networks to computational metal-forming problems. Furthermore, they highlight the network’s potential applicability for accelerated process analysis, inverse parameter identification, and future Digital Twin systems.
- At the current stage, the framework remains limited to a simplified axisymmetric benchmark formulation operating under isothermal assumptions. Future work will focus on extending the methodology toward full three-dimensional blade geometries, thermomechanical coupling, adaptive collocation refinement, and microstructural evolution modeling.

Author Contributions

V.F. and A.M.: conceptualization, methodology, writing—original draft preparation, supervision; V.F.: investigation, validation, visualization; A.M.: resources, project administration. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

Raw data available on request.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

Claude (Anthropic) was used to assist with code development and debugging of the PINN implementation. Google Gemini was used to assist with translation of the manuscript. All scientific content, methodology, results interpretation, and conclusions were formulated and verified by the authors.

References

1. Kobayashi, S.; Oh, S.-I.; Altan, T. *Metal Forming and the Finite-Element Method*; Oxford University Press: New York, NY, USA, 1989; p. 377.
2. Wagoner, R.H.; Chenot, J.L. *Fundamentals of Metal Forming*; John Wiley & Sons: New York, NY, USA, 1996; p. 405.
3. Wu, S.; Zhou, Z.; Bruce, N.; et al. A simulation of Al-Si coating growth under various hot stamping austenitization parameters: An artificial neural network model. *Mater. Today Commun.* **2024**, *38*, 108492. <https://doi.org/10.1016/j.mtcomm.2024.108492>.
4. Zhao, Y.; Li, H.; Zhou, H.; et al. Rapid prediction of material deformation in hot stamping of battery box geometries using graph neural network. *J. Phys. Conf. Ser.* **2025**, *3104*, 012057. <https://doi.org/10.1088/1742-6596/3104/1/012057>.
5. Liu, S.; Xia, Y.; Liu, Y.; et al. Tool path planning of consecutive free-form sheet metal stamping with deep learning. *J. Mater. Process. Technol.* **2022**, *303*, 117530. <https://doi.org/10.1016/j.jmatprotec.2022.117530>.
6. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>.
7. Haghighat, E.; Raissi, M.; Moure, A.; et al. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput. Methods Appl. Mech. Eng.* **2021**, *379*, 113741. <https://doi.org/10.1016/j.cma.2021.113741>.
8. Niu, S.; Zhang, E.; Bazilevs, Y.; et al. Modeling finite-strain plasticity using physics-informed neural network and assessment of the network performance. *J. Mech. Phys. Solids* **2023**, *172*, 105177. <https://doi.org/10.1016/j.jmps.2022.105177>.
9. Ding, F.; Hong, T.; Dong, F.; et al. Research Advancements in High-Temperature Constitutive Models of Metallic Materials. *Crystals* **2025**, *15*, 699. <https://doi.org/10.3390/cryst15080699>.
10. Dornheim, J.; Morand, L.; Nallani, H.J.; et al. Neural Networks for Constitutive Modeling: From Universal Function Approximators to Advanced Models and the Integration of Physics. *Arch. Comput. Methods Eng.* **2024**, *31*, 1097–1127. <https://doi.org/10.1007/s11831-023-10009-y>.
11. Munzone, F.; Hazrati, J.; Hakvoort, W.; et al. Comparative study of artificial neural network and physics-informed neural network application in sheet metal forming. *Mater. Res. Proc.* **2024**, *41*, 2278–2288. <https://doi.org/10.21741/9781644903131-251>.
12. Bobbili, V.S.R.; Chandrasekhara Sastry, C.; Krishnaiah, J.; et al. Physics-informed neural networks for predicting high-strain-rate energy absorption in additively manufactured lattice materials. *Prog. Addit. Manuf.* **2026**, *11*, 2095–2121. <https://doi.org/10.1007/s40964-025-01460-3>.
13. Sun, Y.; Zhang, Q.; Raffoul, S. Physics-informed neural network for predicting hot-rolled steel temperatures during heating process. *J. Eng. Res.* **2025**, *13*, 1496–1504. <https://doi.org/10.1016/j.jer.2024.02.011>.
14. Almeida, E.F.; Silva, S.; Cunha, A., Jr. Physics-informed neural networks for solving elasticity problems. In Proceedings of the 27th International Congress of Mechanical Engineering (COBEM 2023), Florianópolis, Brazil, 4–8 December 2023.
15. Roy, A.M.; Guha, S. A data-driven physics-constrained deep learning computational framework for solving von Mises plasticity. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106049. <https://doi.org/10.1016/j.engappai.2023.106049>.
16. Roy, A.M.; Guha, S.; Sundararaghavan, V.; et al. Physics-infused deep neural network for solution of non-associative Drucker–Prager elastoplastic constitutive model. *J. Mech. Phys. Solids* **2024**, *185*, 105570. <https://doi.org/10.1016/j.jmps.2024.105570>.
17. Galib, G.; Silva, F.J.G.; Pedroso, A.F.V.; et al. A Comprehensive Review of Additive Manufacturing Technologies for Composite Materials. *J. Mech. Eng. Manuf.* **2025**, *1*, 2. <https://doi.org/10.53941/jmem.2025.100002>.
18. Pinto, G.F.; Almeida, D.; Silva, F.J.G.; et al. Correlating Cutting Performance and Surface Roughness under Different Bias Using TiAlTaN Coated Milling Tools. *J. Mech. Eng. Manuf.* **2025**, *1*, 7. <https://doi.org/10.53941/jmem.2025.100007>.
19. Guimarães, A.; Pereira, R.; Pereira, M.; et al. Business Productivity and the Adoption of Lean and Industry 4.0 Tools: A Regional Study. *J. Mech. Eng. Manuf.* **2025**, *1*, 6. <https://doi.org/10.53941/jmem.2025.100006>.

20. Ren, Z.; Zhou, S.; Liu, D.; et al. Physics-Informed Neural Networks: A Review of Methodological Evolution, Theoretical Foundations, and Interdisciplinary Frontiers Toward Next-Generation Scientific Computing. *Appl. Sci.* **2025**, *15*, 8092. <https://doi.org/10.3390/app15148092>.
21. Yang, S.; Kim, H.; Hong, Y.; et al. Data-driven physics-informed neural networks: A digital twin perspective. *Comput. Methods Appl. Mech. Eng.* **2024**, *428*, 117075. <https://doi.org/10.1016/j.cma.2024.117075>.
22. Paszke, A.; Gross, S.; Massa, F.; et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019.
23. Lu, L.; Meng, X.; Mao, Z.; et al. DeepXDE: A deep learning library for solving differential equations. *SIAM Rev.* **2021**, *63*, 208–228. <https://doi.org/10.1137/19M1274067>.
24. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
25. Liu, D.C.; Nocedal, J. On the limited memory method for large scale optimization. *Math. Program.* **1989**, *45*, 503–528. <https://doi.org/10.1007/BF01589116>.
26. Hoff, N.J. Approximate analysis of structures in the presence of moderately large creep deformations. *Q. Appl. Math.* **1954**, *12*, 49–55. <https://doi.org/10.1090/qam/61664>.
27. Norton, F.H. *The Creep of Steel at High Temperatures*; McGraw-Hill: New York, NY, USA, 1929.
28. Tancik, M.; Srinivasan, P.P.; Mildenhall, B.; et al. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems 33*; Larochelle, H., Ranzato, M., Hadsell, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 7537–7547.