
Article

Not peer-reviewed version

A Regularized Physics-Informed Neural Network to support Data-Driven Nonlinear Constrained Optimization

[Diego Armando Perez-Rosero](#) ^{*}, [Andrés Marino Álvarez-Meza](#), [Germán Castellanos-Dominguez](#)

Posted Date: 17 June 2024

doi: [10.20944/preprints202406.1118.v1](https://doi.org/10.20944/preprints202406.1118.v1)

Keywords: Nonlinear optimization; physics-informed neural networks; regularization; data-driven



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

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

Article

A Regularized Physics-Informed Neural Network to Support Data-Driven Nonlinear Constrained Optimization

Diego Armando Perez-Rosero * , Andrés Marino Álvarez-Meza  and Cesar German Castellanos-Dominguez 

Signal Processing and Recognition Group, Universidad Nacional de Colombia, 170003 Manizales, Colombia; dieaperezros@unal.edu.co (D.P.-R.), amalvarezme@unal.edu.co (A.A.-M.), cgcastellanosd@unal.edu.co (G.C.-D.)

* Correspondence: dieaperezros@unal.edu.co

Abstract: Nonlinear optimization (NOPT) is a meaningful tool for solving complex tasks in fields like engineering, economics, and operations research, among others. However, NOPT has problems when it comes to dealing with data variability and noisy input measurements that lead to incorrect solutions. Furthermore, nonlinear constraints may result in outcomes that are either infeasible or suboptimal, such as non-convex optimization. This paper introduces a novel regularized physics-informed neural network (RPINN) framework as a new NOPT tool for both supervised and unsupervised data-driven scenarios. By using custom activation functions and regularization penalties in an artificial neural network (ANN), RPINN can handle data variability and noisy inputs. Besides, it uses physics principles to build the network architecture, computing the optimization variables based on network weights and learned features. In addition, it uses automatic differentiation training to make the system scalable and cut down on computation time through batch-based back-propagation. The test results for both supervised and unsupervised NOPT tasks show that our RPINN can provide solutions that are competitive compared to state-of-the-art solvers. In turn, RPINN's robustness against noisy input measurements makes it particularly valuable in environments with fluctuating information. Additionally, RPINN's ANN-based foundation offers significant flexibility and scalability.

Keywords: nonlinear optimization; physics-informed neural networks; regularization; data-driven

1. Introduction

Optimization approaches have emerged as tools for solving complex problems across various disciplines. Unlike traditional linear models, nonlinear optimization (NOPT) methods are capable of incorporating the intricate and interdependent relationships inherent in real-world scenarios [1]. These techniques are particularly valuable in fields such as engineering, economics, and operations research, where they enable the formulation and solution of models that more accurately reflect the underlying dynamics [2]. By leveraging advanced algorithms and computational solutions, NOPT facilitates improved decision-making and implementation, thereby enhancing efficiency and effectiveness in tackling multifaceted challenges. As research and technology continue to evolve, their significance in achieving optimal outcomes in diverse applications is becoming increasingly evident [3,4]. Nonetheless, NOPT comprises salient issues: First, data variability and noisy input measurements yield erroneous and fluctuating solutions. Second, nonlinear constraints greatly complicate the task of achieving optimal outputs [5]. Moreover, system scalability should be considered.

Data variability and noisy samples, in particular, are known to be a problem that makes stochastic measurements less accurate and increases the number of errors in NOPT [6]. The presence of unwanted effects in the data not only reduces the solution quality but also adds complications to the computation, making it more difficult to choose suitable optimization parameters [7]. The instability greatly impedes the optimization process, rendering the algorithm vulnerable to external effects and significantly reducing its overall efficiency [8]. Besides, the intricacies of nonlinear constraints might result in outcomes that are either infeasible or suboptimal [9]. Then, the NOPT may have a slow rate of convergence, with a tendency to become trapped at a local minimum. This might present a challenge

when both speed and accuracy are crucial [10]. Hence, optimization techniques become impractical for large-scale applications [11], and as the number of variables increases, scalability becomes a significant hindrance, underscoring the pressing need for specialist software and more processing time [12]. Consequently, it is important to deal with large optimization problems, reduce runtime, and simplify the inherent complexity of noisy inputs and nonlinear constraints [11]. Indeed, many NOPT tasks are nondeterministic polynomial-time (NP-hard), making it difficult to find an exact solution for large instances because there is not a polynomial-time algorithm that works well or that does not introduce errors into the final output [13]. Additionally, some NOPT tasks have non-convex nonlinear programming (NLP) issues. The latter are especially challenging because they involve a lot of non-convex and integer functions [14].

Typically, mathematical programming or other classical techniques solve NOPT. These methods are capable of effectively handling non-linearities and discontinuities [9]. Customized strategies are also implemented to refine the iterative search [15]. Gradient-based techniques, mostly based on descent methods, have also shown they can deal with problems like non-linear and convex constraints [16]. Similarly, decomposition methods simplify complexity by segmenting the optimization into more manageable subproblems [17]. Additionally, search approaches and metaheuristics are crucial for maintaining a proper balance between exploration and exploitation [18], which enhances efficiency in finding optimal outputs. However, conventional methods often converge on solutions that may not be useful, especially in stochastic and noisy environments with high uncertainty and intrinsic data variability, which can reduce their accuracy [19].

Nowadays, artificial neural networks (ANNs) employ supervised learning to tackle nonlinear and stochastic problems through regression tasks. These networks are trained to find complex patterns and make accurate predictions even when there is a lot of uncertainty using data-driven strategies [20]. Commonly, ANN-based approaches employ automatic differentiation (AD), a computational technique used to evaluate the derivatives of functions efficiently and accurately. Unlike numerical alternatives, which can suffer from precision issues, or symbolic differentiation, which can be computationally expensive, AD works by breaking down functions into elementary operations for which derivatives are known and applying the chain rule systematically [21]. This process ensures that the derivative calculations are exact to machine precision and enables the calculation of loss function gradients with respect to network parameters, which is essential for gradient-based optimization algorithms like back-propagation.

Recently, physics-informed neural networks (PINNs) have emerged as an effective ANN-based optimization technique. Designed to align training with relevant physical principles, they have proven successful in various NOPT applications [22]. Commonly, the Karush-Kuhn-Tucker (KKT) criteria are used to represent constraints and integrate them into the network's cost function during supervised training [23]. Additionally, a novel approach for integrating constraints using Runge-Kutta (RK) in unsupervised training has been proposed in [24]. Nevertheless, putting these networks into action is hard, especially when it comes to defining the right loss functions, choosing the best hyperparameters, and making sure that computations run quickly while complex systems are being trained [25]. Also, although PINNs have remarkable capabilities, their ability to generalize to nonlinear optimization problems is limited [26].

In this paper, we present a novel regularized PINN framework, termed RPINN, as a NOPT optimization tool for both supervised and unsupervised data-driven scenarios. As a result, we deal with three key NOPT issues. We first address data variability and noisy input measurements by appropriately adapting custom activation and regularization penalties within an ANN scheme. Second, we effectively integrate nonlinear constraints into the network architecture, adhering to the principles of model physics. Specifically, we utilize the network weights and/or learned features within a functional composition framework to determine the NOPT variables. Third, our ANN-based strategy employs AD training, which favors system scalability and computational time through batch-based back-propagation. Experimental results from both supervised and unsupervised data-driven NOPT tasks

confirm that our proposal is robust and competitive against state-of-the-art optimization approaches. The primary advantage of our proposal lies in its stability against noisy input measurements, making it a particularly valuable solution in contexts with fluctuating information. Furthermore, because RPINN is based on ANN, it offers flexibility in terms of network architecture.

The agenda for this paper is as follows: Section 2 summarizes the related work. Section 3 describes the materials and methods. Sections 5 and 6 depict the experiments and discuss the results. Lastly, Section 7 outlines the conclusions and future work.

2. Related Work

Some studies have shown that mathematical programming has become a crucial tool in numerical optimization. A notable example is the analysis by [9], which employs a sequential linear programming algorithm to address nonlinearities and discontinuities. In this context, the simplex method proves essential, being a classic technique effective for solving linear programming problems through iterative adjustments of solutions within a feasible set [27]. Similarly, the study by [15] explores a solution via quadratic programming (QP). Mixed-integer programming (MIP), on the other hand, is an optimization strategy that uses both integer and continuous variables. It is widely used to solve difficult problems [28], focusing on how the branch-and-cut (BC) algorithm can be employed to find the best solution [29]. Furthermore, second-order cone programming (SOCP) facilitates effective solutions for problems involving linear and quadratic constraints [30]. New studies, like [31], look into semidefinite programming (SDP), and the work in [32] uses convexification techniques. Likewise, exponential programming (EXP), which models NOPT objectives and constraints through exponential functions [33]. Additionally, power cone programming (PCP) is considered for modeling product and square relationships [34]. Yet, these classical methods face challenges such as scalability, computation time, convergence, and practical precision, underscoring their inherent complexity and limitations. Furthermore, the use of relaxations or approximations affects the optimization accuracy [35].

On the other hand, gradient methods' efficiency and precision in identifying optimal solutions highlight their relevance for practical optimization tasks. The work in [36] uses the Dai-Liao conjugate gradient method and hyperplane projections for global convergence to solve nonlinear equations. In addition, [37] faces the non-convex issue based on a set of starting points. Moreover, nonlinear decomposition using linear programming (LP) and gradient descent was also proposed [38]. Further, the work in [39] examined the Newton-based search to deal with convergence issues in poorly conditioned systems. Also, the semi-sweeping Newton technique was applied for optimization in Hilbert spaces [40]. For noisy problems, the authors in [41] use piecewise polynomial interpolation and box reformulations, along with an interior-point (IP) method. Authors in [42] tackle similar problems with integrated penalty techniques. Overall, gradient methods are effective at solving NOPT tasks, but they have a challenging time convergent and are expensive to run in noisy and nonlinear situations [43]. Besides, it can be challenging to choose the best learning rate, and they run the risk of finding local minima [44]. As seen in [45], it is also important to make sure that at least first-degree differentiation continuity is maintained when using techniques like the conjugate gradient, the IP, and the Newton-based approach.

Of note, most of the available optimization solvers are based on the classical approaches mentioned above. Among them, Clarabel stands out for its versatility in optimizing a wide variety of problems. However, it still faces significant challenges in areas such as MIP [46]. Gurobi is renowned for its proficiency in MIP due to its extensive range of techniques, including simplex and IP methods. However, because it is proprietary software, it might not be able to be used in situations that require license flexibility [47]. Mosek is efficient concerning the IP approach, but its support for MIP is relatively limited, and its aptitude for NLP remains under debate, which could be a hindrance for developers who prefer open-source solutions [48]. Xpress specializes in solving MIP, offers conditional support for NLP, but is a closed-license alternative [49]. In turn, SCS, leveraging its open-source status, promotes adaptability and collaborative development, although its limitations in NLP reduce its effectiveness in

certain optimization areas [50]. IPOPT excels at solving NLP problems, and its open access allows for flexibility [51].

Now, in this multifaceted optimization environment, the integration of tools such as MATPOWER, GEKKO, and CVXPY significantly expands the available options. MATPOWER is essential for solving energy system issues and supports solvers like Gurobi, Xpress, and IPOPT for linear, mixed-integer, and nonlinear programming [52–54]. GEKKO specializes in dynamic systems and nonlinear models, offering a holistic and open-source Python platform [55,56]. CVXPY is an open-source modeling language for convex optimization problems embedded in Python. It allows you to express your problems naturally, mirroring the mathematical formulation rather than conforming to the restrictive standard form required by solvers [57,58]. Table 1 summarizes the mentioned solvers.

Table 1. State-of-the-art solvers for optimization. (*) Except mixed-integer SDP. (**) Features available with the licensed version only.

Solver	LP	QP	SOCP	SDP	EXP	PCP	MIP	NLP	Strategy	Open source	Software
Clarabel [46]	✓	✓	✓	✓	✓	x	x	x	IP	✓	CVXPY
Gurobi [47]	✓	✓	✓	x	x	x	✓	x	IP, Simplex, BC	x	MATPOWER, CVXPY
Mosek [48]	✓	✓	✓	✓	✓	✓	✓*	x	IP	x	MATPOWER, CVXPY
Xpress [49]	✓	✓	✓	x	x	x	✓	✓**	IP, Simplex, BC	x	CVXPY
SCS [50,59]	✓	✓	✓	✓	✓	✓	x	x	IP	✓	CVXPY
IPOPT [51]	✓	✓	✓	✓	✓	✓	✓	✓	IP	✓	MATPOWER, GEKKO

Recently, ANNs have positioned themselves as fundamental tools in optimization by incorporating deep learning techniques, effectively addressing the complexity and non-linearities of various problems. Conventional ANNs employ supervised learning to tackle nonlinear and stochastic problems through regression tasks. To this end, historical data or solutions precomputed by specialized NOPT tools are used to train these networks [60]. This approach enables ANNs to learn complex patterns and make accurate predictions even under significant uncertainty [20]. Typically, ANN-based approaches utilize AD, a computational method for efficiently and accurately evaluating function derivatives. Instead of numerical or symbolic differentiation, which can have issues with accuracy and require a lot of computing power, AD breaks functions down into simple operations whose derivatives are known and uses the chain rule consistently [21]. Thereby, AD ensures machine-level accuracy in derivative calculations and simplifies the determination of loss function gradients in relation to network parameters, enabling the use of gradient-based search with back-propagation. The work in [61] combines quasi-Newton methods and ANNs for NOPT. Furthermore, the authors in [60] utilize deep learning to solve optimal flow problems. Similarly, the work in [62] introduces an integrated training technique that, while effective, requires larger neural networks and presents challenges in generalization. Concurrently, [63] uses elastic layers and incremental training as optimization-based solvers. Furthermore, the method by [64] combines convex relaxation with graph neural networks.

Besides, PINN has recently emerged as a powerful optimization tool. These training approaches have proven effective in various NOPT applications, integrating relevant physical principles within ANNs [22]. The KKT criteria are applied to formulate constraints that are incorporated into an ANN's cost function during supervised training [23]. In [65], a PINN framework is detailed that imposes penalties for constraint violations in the loss function. The study in [66] proposes a loss function that combines errors from differential and algebraic states with normative equation violations. Additionally, a novel strategy has been proposed to include constraints in unsupervised training using an RK-based technique [24]. Nevertheless, complete approaches based on ANNs and PINNs face challenges such as optimality degradation. In response, advanced alternatives like [67] have emerged, integrating system constraints into the cost function and applying penalties for violations. Furthermore, [68] introduces an algorithm to address nonlinear problems modeled by partial differential equations with noisy data through Bayesian physics-informed neural networks (B-PINNs). Additionally, [69] proposes a parametric differential equation-based approach holding functional connections to enhance the robustness and accuracy of PINNs. In turn, [70] presents a truncated Fourier decomposition,

termed Modal-PINNs, to optimize the reconstruction of periodic signals. However, these alternatives often lack adequate precision, generalization capability, and scalability [71]. Finally, supervised data is usually required, complicating their application in various NOPT scenarios.

3. Materials and Methods

3.1. Nonlinear Optimization Fundamentals (NOPT)

Let $\mathbf{x} \in \mathbb{R}^P$ be a vector in P variables. The conventional NOPT problem can be summarized as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \varrho(\mathbf{x}) \\ \text{s.t.} \quad & \xi_{\min} \leq \mathbf{x} \leq \xi_{\max} \\ & h_L(\mathbf{x}) \leq \mathbf{0} \\ & h_N(\mathbf{x}) \leq \mathbf{0}, \end{aligned} \quad (1)$$

where the objective function $\varrho : \mathbb{R}^P \rightarrow \mathbb{R}$ is real-valued. Also, the bound constraints are shown by $\xi_{\min}, \xi_{\max} \in \mathbb{R}^P$. The linear and nonlinear constraints are described by $h_L : \mathbb{R}^P \rightarrow \mathbb{R}^{C_L}$ and $h_N : \mathbb{R}^P \rightarrow \mathbb{R}^{C_N}$, where $C_L \in \mathbb{N}$ and $C_N \in \mathbb{N}$.

Figure 1 depicts the main pipeline of the classical approaches for NOPT. First, it includes the physical system's parameters, constraints, limits, and the objective function to be optimized. Second, starting from an initial point, the optimization algorithm iterates until convergence. Of note, the number of iterations, the level of improvement, and the objective function thresholding are the relevant stopping criteria to return the final output.

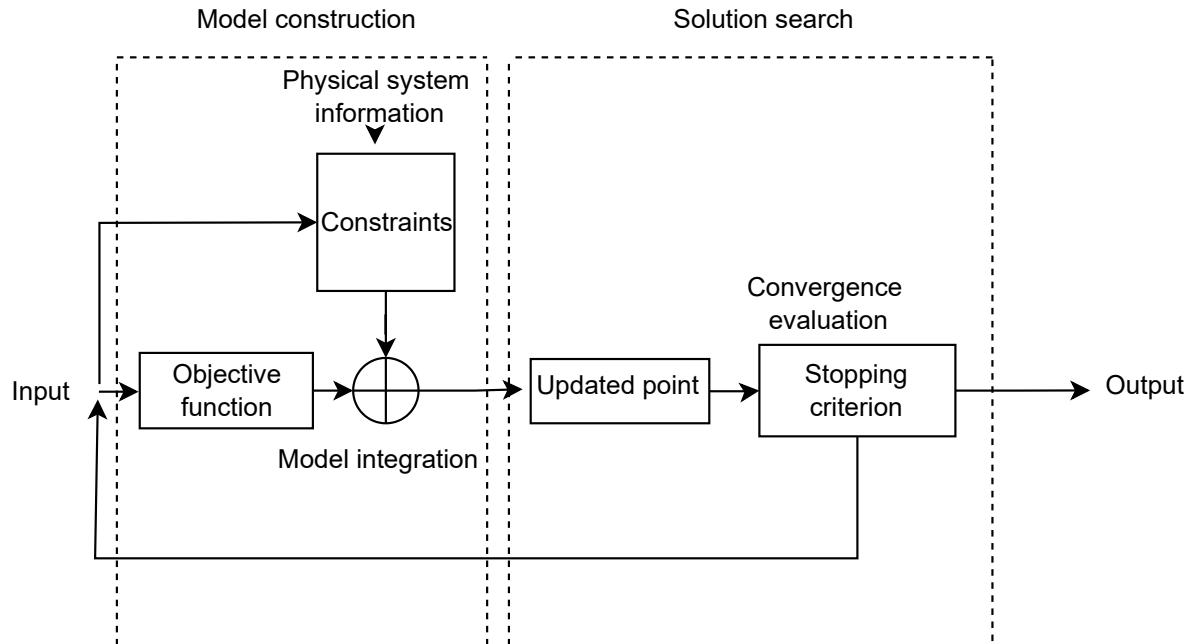


Figure 1. Classical optimization pipeline for NOPT.

3.2. Regularized Physics-Informed Neural Network (RPINN)

Let $\{y_r \in \mathcal{Y}, z_r \in \mathcal{Z}\}_{r=1}^R$ be an input-output set holding R samples. Our data-driven RPINN approach aims to couple the optimization problem in Eq. 1 as a penalty-based loss with bounded constraints from both network weights and learned features, as follows:

$$\begin{aligned}
 & \min_{\tilde{\mathbf{X}}, \tilde{\mathbf{Z}}} \frac{\lambda_{\mathcal{L}}}{R} \sum_{r=1}^R \mathcal{L}(y_r, \tilde{f}(z_r | \tilde{\mathbf{X}}, \tilde{\mathbf{Z}})) + \sum_{i=1}^{C_L} \frac{\lambda_{Li}}{R} \sum_{r=1}^R \tilde{h}_{Li}(y_r, \tilde{f}(z_r | \tilde{\mathbf{X}}, \tilde{\mathbf{Z}})) + \sum_{j=1}^{C_N} \frac{\lambda_{Nj}}{R} \sum_{r=1}^R \tilde{h}_{Nj}(y_r, \tilde{f}(z_r | \tilde{\mathbf{X}}, \tilde{\mathbf{Z}})) \\
 & \text{s.t. } \lambda_{\mathcal{L}} + \sum_{i=1}^{C_L} \lambda_{Li} + \sum_{j=1}^{C_N} \lambda_{Nj} = 1 \\
 & \zeta_{\min} \leq \tilde{\mathbf{X}} \leq \zeta_{\max} \\
 & \psi_{\min} \leq \tilde{f}(z_r | \tilde{\mathbf{X}}, \tilde{\mathbf{Z}}) \leq \psi_{\max}, \quad \forall r \in R;
 \end{aligned} \tag{2}$$

where $\tilde{f} : \mathcal{Z} \rightarrow \mathcal{Y}$ is an ANN-based mapping function, $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a given loss, $\tilde{\mathbf{X}}$ holds the network parameters, and $\tilde{\mathbf{Z}}$ gathers the learned features along layers. Also, $\tilde{h}_{Li}(\cdot, \cdot)$ and $\tilde{h}_{Nj}(\cdot, \cdot)$ are the i -th linear and j -th nonlinear penalty functions to follow the NOPT constraints set by the regularization terms $\lambda_{\mathcal{L}}, \lambda_{Li}, \lambda_{Nj} \in [0, 1]$, where $i \in \{1, 2, \dots, C_L\}$ and $j \in \{1, 2, \dots, C_N\}$. Furthermore, ζ_{\min} and ζ_{\max} collect the network parameter limit values, and ψ_{\min} and ψ_{\max} capture the network output and feature bounds.

For a given input $z \in \mathcal{Z}$, our deep learning-based function with \hat{L} layers yields:

$$\begin{aligned}
 \tilde{f}(z | \tilde{\mathbf{X}}, \tilde{\mathbf{Z}}) &= (f_{\hat{L}} \circ \dots \circ f_1 | \tilde{\mathbf{X}}, \tilde{\mathbf{Z}})(z), \\
 \tilde{\mathbf{z}}_l &= f_l(\tilde{\mathbf{z}}_{l-1} | \tilde{\mathbf{x}}_l, b_l) = \nu_l(\tilde{\mathbf{x}}_l^\top \tilde{\mathbf{z}}_{l-1} + b_l).
 \end{aligned} \tag{3}$$

In the l -th layer of Eq. 3, where $l \in \{1, 2, \dots, \hat{L}\}$, the weights and bias are $\tilde{\mathbf{x}}_l, b_l \in \tilde{\mathbf{X}}$, the learned feature vector is $\tilde{\mathbf{z}}_l \in \tilde{\mathbf{Z}}$, and $\nu_l(\cdot)$ is a nonlinear activation function to deal with both network representation and customized bounds to fulfill the Eq. 2 limit constraints. Furthermore, the RINN optimization problem can be solved via gradient descent with AD and back-propagation [72].

It is worth noting that our baseline RINN studies a supervised scenario for simplicity, but by addressing its regularized loss, we can easily achieve an unsupervised extension. Figure 2 depicts the RINN main sketch.

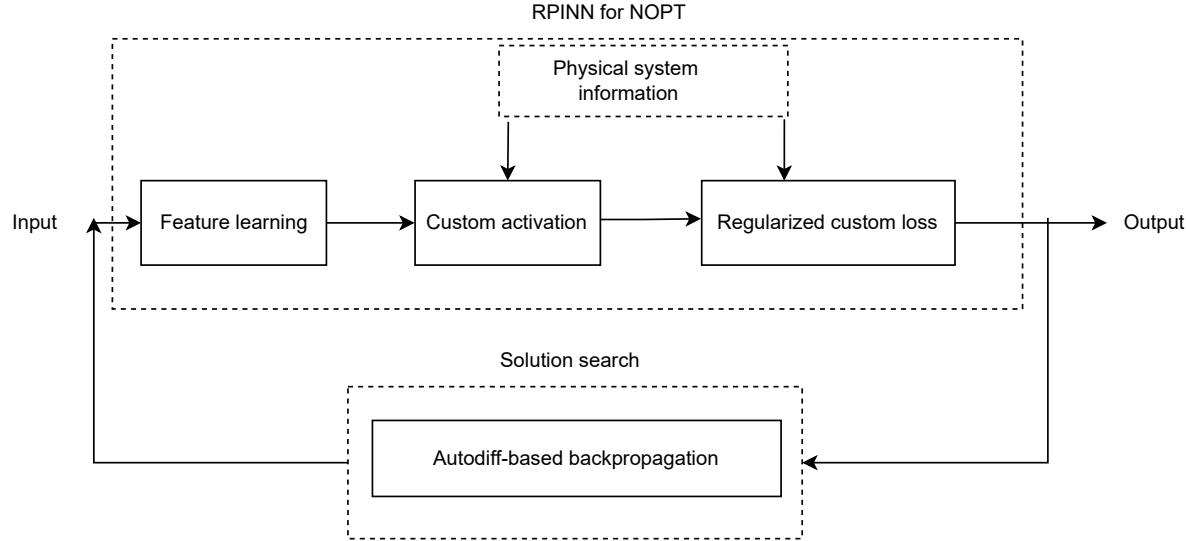


Figure 2. Regularized physics-informed neural network for data-driven nonlinear constrained optimization main sketch .

4. Tested Scenarios for NOPT Using RPINN

We study two main datasets to test our RPINN as a data-driven NOPT approach: i) a constrained uniform mixture model with nonlinear loss and supervised target; and ii) a constrained flow and

pressure gas-powered system optimization with unsupervised loss. Below, we provide a detailed description of each experiment.

4.1. Supervised Constrained Optimization: Uniform Mixture Model

This task comprises a linear and bound-constrained optimization of a nonlinear cost [73]:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{r=1}^R |y_r - \mathbf{x}^\top \mathbf{z}_r|_2^2 \\ \text{s.t.} \quad & \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}, \\ & \mathbf{x}^\top \mathbf{1} = 1; \end{aligned} \quad (4)$$

where $y_r \in \mathbb{R}^+$ is the r -th target output, $\mathbf{x} \in \mathbb{R}^P$ denotes the mixing coefficients, and $\mathbf{z}_r \in \mathbb{R}^P$ holds random samples drawn from a uniform distribution as: $z_{rp} \sim \mathcal{U}(z|p-1, p)$. $\mathbf{0}$ and $\mathbf{1}$ are all-zeros and all-ones vectors of proper size.

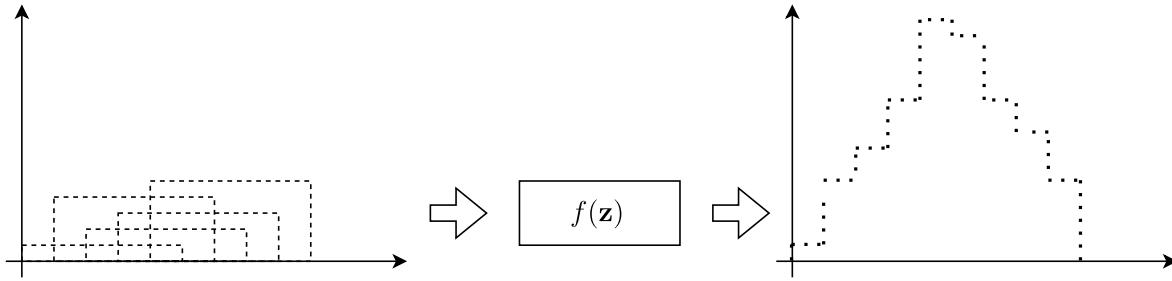


Figure 3. Uniform mixture model optimization. Left: weighted uniform probabilities. Right: visual representation of the mixing results.

The optimization problem in Eq. 4 can be solved through our RPINN as follows:

$$\begin{aligned} \min_{\tilde{\mathbf{x}}} \quad & \frac{\lambda_L}{R} \sum_{r=1}^R \mathcal{L}_H(y_r, \tilde{f}(\mathbf{z}_r | \tilde{\mathbf{X}}); \epsilon) + \frac{\lambda_L}{R} \tilde{\mathbf{x}}_L^\top \mathbf{1} \\ \text{s.t.} \quad & \lambda_L + \lambda_L = 1, \quad \forall \lambda_L, \lambda_L \in [0, 1] \\ & \mathbf{0} \leq \tilde{\mathbf{x}}_L \leq \mathbf{1}. \end{aligned} \quad (5)$$

For concrete testing and to mitigate noisy samples, a Huber-based loss is used in Eq. 5:

$$\mathcal{L}_H(y, \tilde{f}(\mathbf{z} | \tilde{\mathbf{X}}); \epsilon) = \begin{cases} \frac{1}{2} \|y - \tilde{f}(\mathbf{z} | \tilde{\mathbf{X}})\|^2 & \|y - \tilde{f}(\mathbf{z} | \tilde{\mathbf{X}})\| \leq \epsilon \\ \epsilon \cdot (\|y - \tilde{f}(\mathbf{z} | \tilde{\mathbf{X}})\| - \frac{1}{2}\epsilon) & \|y - \tilde{f}(\mathbf{z} | \tilde{\mathbf{X}})\| > \epsilon, \end{cases} \quad (6)$$

where $\epsilon \in \mathbb{R}^+$. Next, we fix a scaled exponential linear (SELU) activation for the network function composition, as follows:

$$\text{SELU}(x) = \begin{cases} \theta x & x > 0 \\ \theta \vartheta \cdot (e^x - 1) & x \leq 0, \end{cases} \quad (7)$$

where $\theta, \vartheta \in \mathbb{R}$. Then, $\tilde{\mathbf{x}}_L$. To fulfill the former NOPT limit restriction, RPINN's weights at the output layer \hat{L} hold a l_1 -based max-constraint.

4.2. Unsupervised Constrained Optimization: Gas-Powered System

We study a gas-powered system as a function of flow and pressure. For this purpose, a synthetic network of eight nodes is used, as detailed in [74] and illustrated in Figure 4.

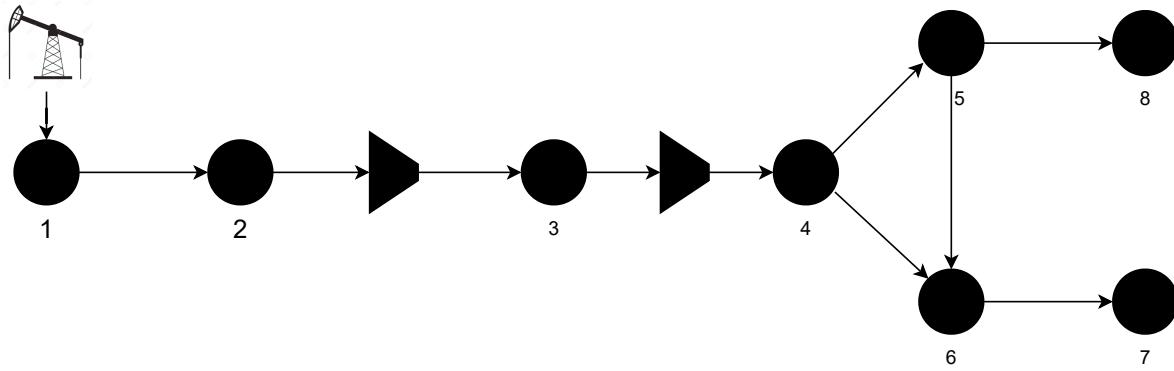


Figure 4. Optimizing gas-powered systems. An eight-node gas network is studied. The diagram depicts the nodes as points, and the arrows indicate flow direction. The trapezoidal shapes represent the pressure compressors.

In particular, the NOPT problem is written as:

$$\begin{aligned}
 \min_{\mathbf{x}, \boldsymbol{\pi}} \quad & \mathbf{x}^\top \mathbf{a} \\
 \text{s.t.} \quad & \mathbf{Bx} = \mathbf{z} \\
 & x_q = \text{sgn}\left(\pi_{w(q)}^2 - \pi_{w'(q)}^2\right) \sqrt{k_q |\pi_{w(q)}^2 - \pi_{w'(q)}^2|}, \\
 & \forall q \in Q; \quad w(q), w'(q) \in W \\
 & \beta_{\min}(n, n') \leq \frac{\pi_n}{\pi_{n'}} \leq \beta_{\max}(n, n'), \quad \forall n, n' \in V \\
 & \gamma_{\min} \leq \boldsymbol{\pi} \leq \gamma_{\max} \\
 & \delta_{\min} \leq \mathbf{x} \leq \delta_{\max},
 \end{aligned} \tag{8}$$

where $\mathbf{a} \in \mathbb{R}^P$ represents the gas transport costs for the P flows in $\mathbf{x} \in \mathbb{R}^P$. The incidence matrix $\mathbf{B} \in \mathbb{R}^{W \times P}$ encodes the gas network structure, with W nodes and $\mathbf{z} \in \mathbb{R}^W$ the input gas demand. The first equality constraint is what encodes the linear-based flow and gas demand equilibrium along the network nodes. Next, the node pressure is stored in $\boldsymbol{\pi} \in \mathbb{R}^W$. In turn, the q -th flow $x_q \in \mathbf{x}$ is selected according to the network structure from \mathbf{B} to fulfill the Weymouth equality with $k_q \in \mathbb{R}$ and $Q \leq P$ [54]. Then, the function $w(q)$ extracts the related pressure $\pi_{w(q)} \in \boldsymbol{\pi}$ regarding such a Weymouth-based physic constraint. Furthermore, $\pi_n, \pi_{n'} \in \boldsymbol{\pi}$ chose the inlet and outlet pressures to fulfill the system compression ratio, with V components ($n, n' \in \{1, 2, \dots, V\}, V \leq W$) and compression factor limits $\beta_{\min}(n, n'), \beta_{\max}(n, n') \in \mathbb{R}^+$. Also, $\gamma_{\min}, \gamma_{\max} \in \mathbb{R}^W$ and $\delta_{\min}, \delta_{\max} \in \mathbb{R}^P$ are the minimum and maximum pressure and flow limits, respectively.

Now, let $\{\mathbf{z}_r \in \mathbb{R}^W\}_{r=1}^R$, be an unsupervised input set concerning the required gas demand for R observations. Our RPINN solution of Eq. 8 is as follows:

$$\begin{aligned}
 \min_{\{\tilde{\mathbf{z}}_r, \tilde{\boldsymbol{\pi}}_r\}_{r=1}^R} \quad & \frac{\lambda_{\mathcal{L}}}{R} \sum_{r=1}^R \tilde{\mathbf{z}}_r^\top \mathbf{a} + \frac{\lambda_{L1}}{R} \sum_{r=1}^R \tilde{h}_{L1}(\mathbf{B}\tilde{\mathbf{z}}_r, \mathbf{z}_r; \epsilon_{L1}) + \\
 & \frac{\lambda_{N1}}{R} \sum_{r=1}^R \tilde{h}_{N1}(\tilde{\mathbf{z}}_r, \tilde{\boldsymbol{\pi}}_r; \mathbf{B}, \epsilon_{N1}) + \frac{\lambda_{N2}}{R} \sum_{r=1}^R \tilde{h}_{N2}(\tilde{\boldsymbol{\pi}}_r; \mathbf{B}, \beta, \epsilon_{N2}) \\
 \text{s.t.} \quad & \lambda_{\mathcal{L}} + \lambda_{L1} + \lambda_{N1} + \lambda_{N2} = 1 \\
 & \tilde{f} = \tilde{f}^\dagger \cup \tilde{f}^\ddagger \\
 & \tilde{\mathbf{z}}_r = \tilde{f}^\dagger(\mathbf{z}_r | \tilde{\mathbf{X}}^\dagger, \tilde{\mathbf{Z}}^\dagger), \tilde{\boldsymbol{\pi}}_r = \tilde{f}^\ddagger(\mathbf{z}_r | \tilde{\mathbf{X}}^\ddagger, \tilde{\mathbf{Z}}^\ddagger) \\
 & \gamma_{\min} \leq \tilde{\boldsymbol{\pi}}_r \leq \gamma_{\max} \\
 & \delta_{\min} \leq \tilde{\mathbf{z}}_r \leq \delta_{\max}, \quad \forall r \in R.
 \end{aligned} \tag{9}$$

Given the r -th gas demand vector $\mathbf{z}_r \in \mathbb{R}^W$, $\tilde{\mathbf{z}}_r \in \mathbb{R}^P$ predicts the flow vector based on f^\dagger , and $\tilde{\boldsymbol{\pi}}_r \in \mathbb{R}^W$ the corresponding pressure vector using f^\dagger . Moreover:

$$\begin{aligned}\tilde{h}_{L1}(\mathbf{B}\tilde{\mathbf{z}}_r, \mathbf{z}_r; \epsilon_{L1}) &= \mathcal{L}_H(\mathbf{B}\tilde{\mathbf{z}}_r, \mathbf{z}_r; \epsilon_{L1}) \\ \tilde{h}_{N1}(\tilde{\mathbf{z}}_r, \tilde{\boldsymbol{\pi}}_r; \mathbf{B}, \epsilon_{N1}) &= \frac{1}{Q} \sum_{q=1}^Q \mathcal{L}_H(\tilde{z}_{rq}, \varphi_q(\tilde{\boldsymbol{\pi}}_r; \mathbf{B}); \epsilon_{N1}) \\ \tilde{h}_{N2}(\tilde{\boldsymbol{\pi}}_r; \mathbf{B}, \boldsymbol{\beta}) &= \frac{1}{V^2} \sum_{n, n' \in V} \mathcal{L}_{\tilde{H}}(\tilde{\pi}_{rn}, \tilde{\pi}_{rn'}; \mathbf{B}, \boldsymbol{\beta}, \epsilon_{N2}),\end{aligned}\quad (10)$$

where notation $\mathcal{L}_H(\cdot, \cdot; \epsilon)$ stands for a Huber-based penalty (see Eq. 6), $\varphi(\tilde{\boldsymbol{\pi}}_r; \mathbf{B}) \in \mathbb{R}^Q$ holds elements:

$$\varphi_q(\tilde{\boldsymbol{\pi}}_r; \mathbf{B}) = \text{sgn}\left(\tilde{\pi}_{rw(q)}^2 - \tilde{\pi}_{rw'(q)}^2\right) \sqrt{k_q |\tilde{\pi}_{rw(q)}^2 - \tilde{\pi}_{rw'(q)}^2|}, \quad \forall q \in Q, \quad (11)$$

and:

$$\mathcal{L}_{\tilde{H}}(\tilde{\pi}_{rn}, \tilde{\pi}_{rn'}; \mathbf{B}, \boldsymbol{\beta}, \epsilon_{N2}) = \begin{cases} 0 & \beta_{\min}(n, n') \leq \frac{\tilde{\pi}_{rn}}{\tilde{\pi}_{rn'}} \leq \beta_{\max}(n, n') \\ \left(\frac{\frac{\tilde{\pi}_{rn}}{\tilde{\pi}_{rn'}} - 0.5\beta_{\min}(n, n')}{\beta_{\max}(n, n')}\right)^2 & \text{otherwise.} \end{cases} \quad (12)$$

It is worth mentioning that the custom penalty in Eq. 10 aims to deal with noisy inputs while preserving the NOPT limits and constraints. In particular, $\mathcal{L}_{\tilde{H}}(\cdot, \cdot; \mathbf{B}, \boldsymbol{\beta}, \epsilon_{N2})$ penalizes pressures that are far from the middle of the compression factor range, according to $\beta_{\min}(n, n'), \beta_{\max}(n, n') \in \boldsymbol{\beta}$. Finally, a scaled sigmoid function $\tilde{\sigma}(\cdot) \in [u_{\min}, u_{\max}]$ addresses the predicted flow and pressure limits in Eq. 9, as:

$$\tilde{\sigma}(x) = \alpha \frac{1}{1 + e^{-x}} + \iota, \quad (13)$$

where $\alpha, \iota \in \mathbb{R}$.

5. Experimental Set-Up

The scenarios in Section 4 will be used to test our RPINN in both supervised and unsupervised settings. They will be utilized to look at sample variability, noisy input measurements, and nonlinear constraints.

5.1. Deep Learning Architectures

To address the uniform mixture model NOPT (supervised constrained optimization), our RPINN consists of two dense layers, as shown in Figure 5 and Table 2.

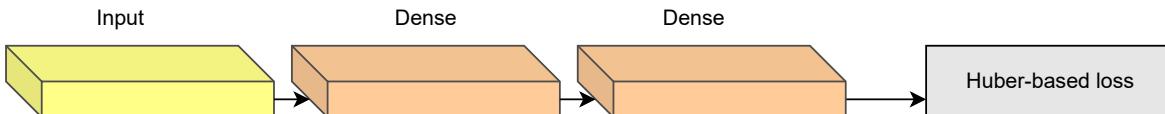


Figure 5. RPINN pipeline for the uniform mixture model-based NOPT.

Table 2. RPINN details for the uniform mixture model-based NOPT. \tilde{R} : batch-size for AD-based back-propagation. Param. #: number of trainable parameters. Total # of parameters: 30.

Layer name	Type	Output shape	Param. #
Input	InputLayer	$(\tilde{R}, 5)$	0
Dense_1	Dense(SELU)	$(\tilde{R}, 5)$	25
Dense_2	Dense(SELU, $l1$ -max-constraint)	$(\tilde{R}, 1)$	5

Next, as seen in Figure 6 and Table 3, a wide ANN architecture is proposed for our RPINN-based gas-powered system scenario. We can specifically focus on essential variables—flows and pressures—in our sketch, adapting it to the unique characteristics of the gas network. To achieve this, our model incorporates blocks of dense layers designed to map input data, as well as batch normalization layers that help stabilize and normalize the features and gradient along the back-propagation. Additionally, it includes custom layers named: custom dense, bounded dense, source switching, and unsupply gas switching. We design these to encode the source behavior of the system, manage unmet demand, and delineate system boundaries.

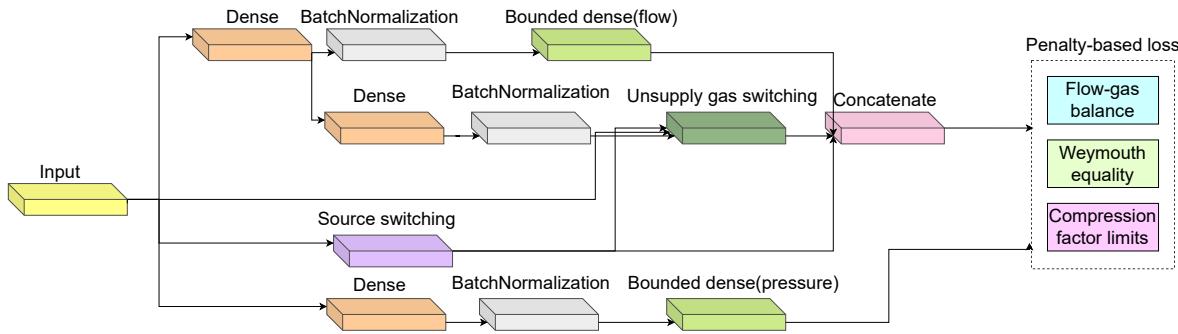


Figure 6. RPINN pipeline for the gas-powered system-based NOPT.

Table 3. RPINN architecture details for the gas-powered system NOPT. \tilde{R} : batch-size for AD-based back-propagation. Source switching, unsupply gas switching, custom dense, and bounded dense stand for specific switching, limited, and scaled layers, as explained in Section 4.2. Param. #: number of trainable parameters. Total # of parameters: 11855.

Layer name	Type	Output shape	Param. #
Input	InputLayer	$(\tilde{R}, 8)$	0
Dense_1	Dense(SELU)	$(\tilde{R}, 236)$	2124
Dense_2	Dense(SELU)	$(\tilde{R}, 8)$	1896
Source switching	CustomDense	$(\tilde{R}, 1)$	1
BatchNormalization_1	BatchNormalization	$(\tilde{R}, 236)$	944
BatchNormalization_2	BatchNormalization	$(\tilde{R}, 8)$	32
Partial flows	BoundedDense	$(\tilde{R}, 50)$	1422
Unsupply gas switching	CustomDense	$(\tilde{R}, 8)$	0
Flow prediction	Concatenate	$(\tilde{R}, 59)$	0
Dense_3	Dense(SELU)	$(\tilde{R}, 236)$	2124
BatchNormalization_3	BatchNormalization	$(\tilde{R}, 236)$	944
Pressure prediction	BoundedDense	$(\tilde{R}, 8)$	1896
Node balance	CustomDense	$(\tilde{R}, 8)$	472
Weymouth	CustomDense	$(\tilde{R}, 14)$	0

5.2. Training Details and Method Comparison

To evaluate the effectiveness of our methodology in addressing optimization problems, we utilized the mean absolute percentage error (MAPE) as the primary performance measure across all conducted experiments, defined as:

$$\text{MAPE}(\tilde{y}_r, \hat{y}_r) = \frac{100}{R} \sum_{r=1}^R \left| \frac{\tilde{y}_r - \hat{y}_r}{\tilde{y}_r} \right| [\%], \quad (14)$$

where $\tilde{y}_r, \hat{y}_r \in \mathbb{R}$ stands for r -th target and predicted value, $\text{MAPE}(\cdot, \cdot) \in [0, 100][\%]$, and $|\cdot|$ is the absolute value operator.

Now, for the uniform mixture model, we generate 500 samples, each composed of five variables. We trained our RPINN architectures on a total of 400 samples, allocating 30% for the validation phase. We used the remaining 100 samples to evaluate the model's performance. To see how well NOPT works with noisy inputs, we add white Gaussian noise to the model output while keeping the signal-to-noise ratio (SNR) value within the set $\{-1, 3, 5\}$. Further, for the gas-powered system, we defined three distinct scenarios to evaluate the network's capacity under varying demand conditions. This process yielded a total of 20.000 samples, of which 30% were designated for testing. We produced 320 samples using GEKKO v1.0.6 to compare the model's performance with IPOPT v3.12 [75].

We implemented RPINN using Python 3.10.12 and the TensorFlow API 2.15.0 on Google Colaboratory. For training, we fixed 500 epochs, a batch size of 64 samples, an Adam optimizer, and a learning rate value of $1e - 3$. The regularization hyperparameters, namely λ . in Eq. 2, are experimentally set using a gridsearch strategy within the set $\{0, 0.25, 0.5, 0.75, 1\}$. Since IPOPT excels at solving NOPT, not to mention its open access, we fix it as a method comparison [51]. Our codes and studied datasets are publicly available at <https://github.com/UN-GCPDS/python-gcpds.optimization> (accessed on 1 March 2024).

6. Results and Discussion

6.1. Supervised Constrained Optimization Results

As shown in Figure 7 (left) and Figure 8, for noisy-free data on the uniform mixture model scenario, both our proposal and the IPOPT solution exhibit similar results. The similarity of the results stems from the fact that the problem defined in Eq. 4 is convex. Next, for noisy inputs, our RPINN, based on the Huber loss function, shows greater robustness against data variability and noise issues. In fact, the Huber function applies the $l1$ -norm for errors exceeding a defined threshold, reducing sensitivity to extreme values, while for smaller errors it uses the $l2$ -norm, ensuring accuracy by penalizing smaller errors. In contrast, the classical IPOPT technique uses an objective function based on the $l2$ -norm, which is sensitive to outliers because it significantly penalizes large deviations. The weight distributions provide support for the latter hypothesis. Noise-free data leads to similar strength predictions for both RPINN and IPOPT. Conversely, for noisy inputs, our proposal regularizes the network weights, yielding concentrated values to find the main output dynamics, and outperforms the IPOPT regarding the MAPE for all considered SNR values.

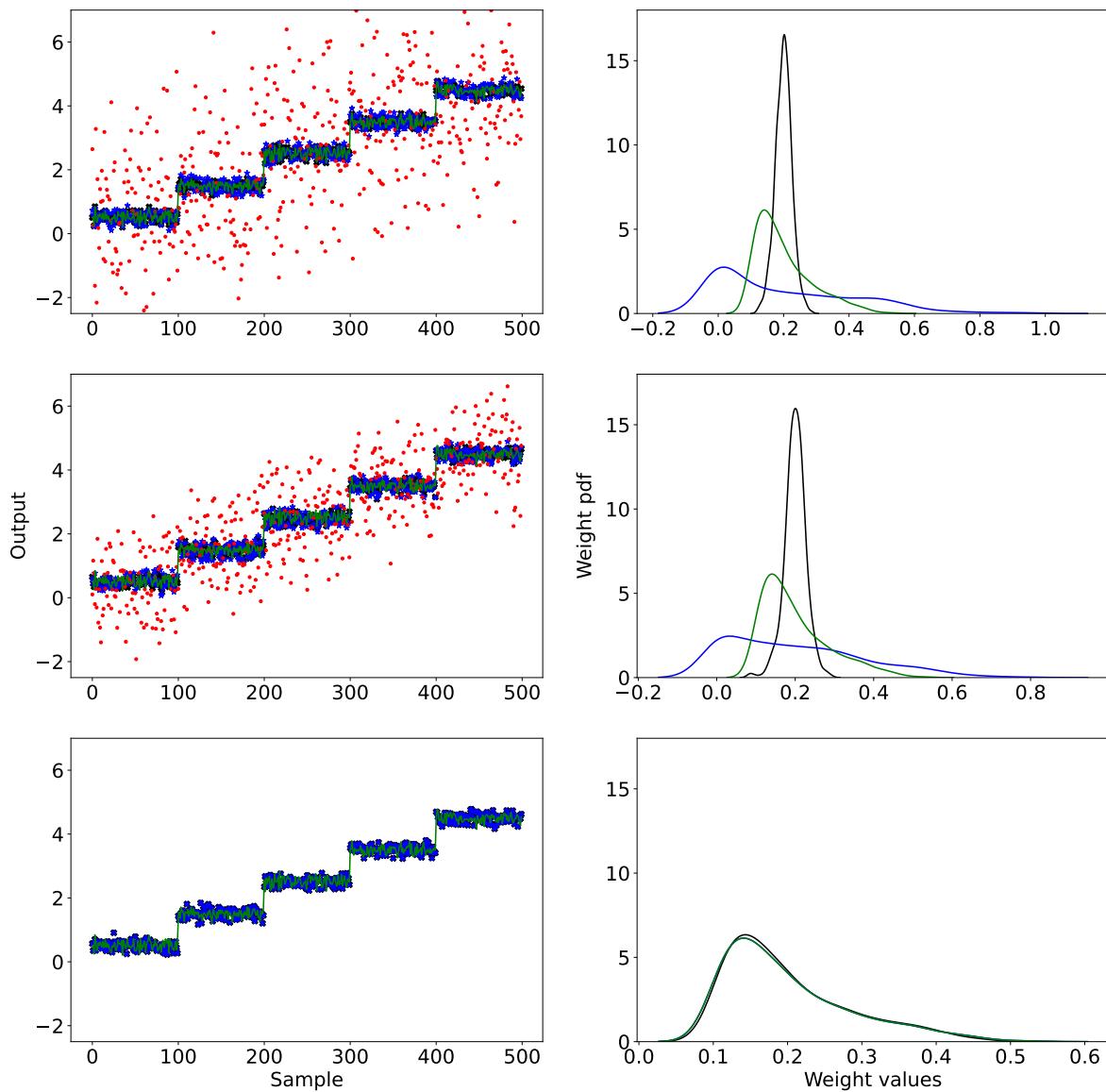


Figure 7. RPINN uniform mixture model-based NOPT results. First row: SNR = -1. Second row: SNR = 3. Third row: noise-free. Left: output prediction. Right: weight distribution. Green: target. Red: noisy target. Black: RPINN. Blue: IPOPT.

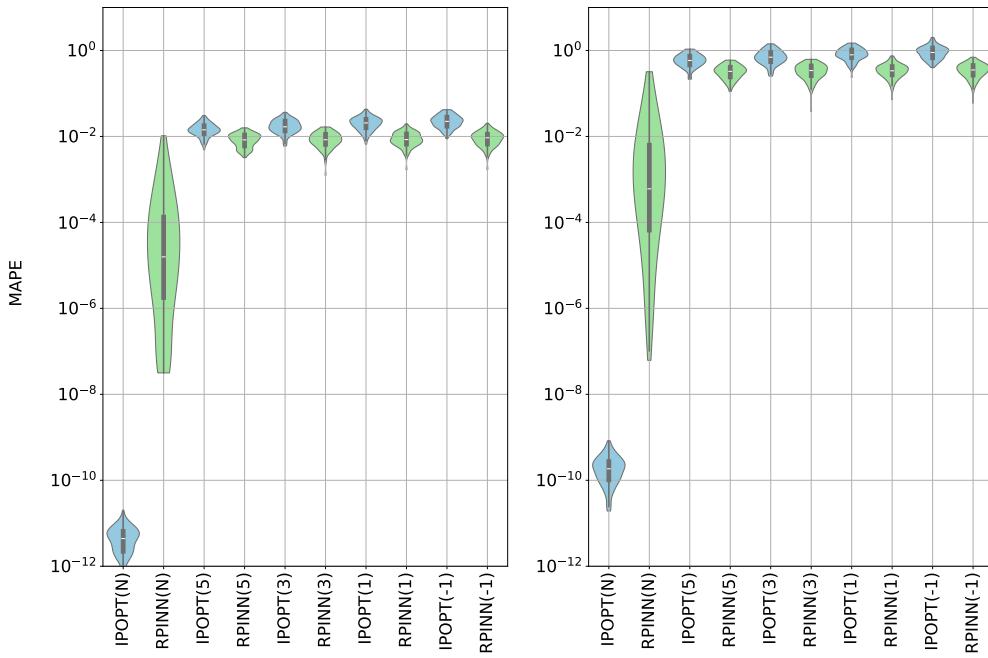


Figure 8. Uniform mixture model MAPE results. Left: output error. Right: weights error. (N): noisy-free. (-1),(3), and (5) stand for the SNR value.

6.2. Unsupervised Constrained Optimization Results

Figure 9 depicts our RPINN regularized penalty illustration for the gas-powered system NOPT. We adopted a standard variant of the Huber loss for the node balance and Weymouth constraints. As shown, the threshold ϵ transitions between the l_1 and l_2 norms. Regarding the constraint on the compression ratio limit, it was essential to alter the structure due to its inequality behavior. This enhancement stabilizes the transition between the l_2 and l_1 norms at zero, based on the distance to the central value of the required range. Furthermore, it is crucial to correctly integrate these cost functions into our RPINN. Then, the right plot in Figure 9 shows the Weymouth (blue) and compression ratio (orange) penalty evolution. As seen, the obtained loss shows a decreasing trend, indicating that the Huber-based approach can handle the physical limitations of the gas-powered NOPT.

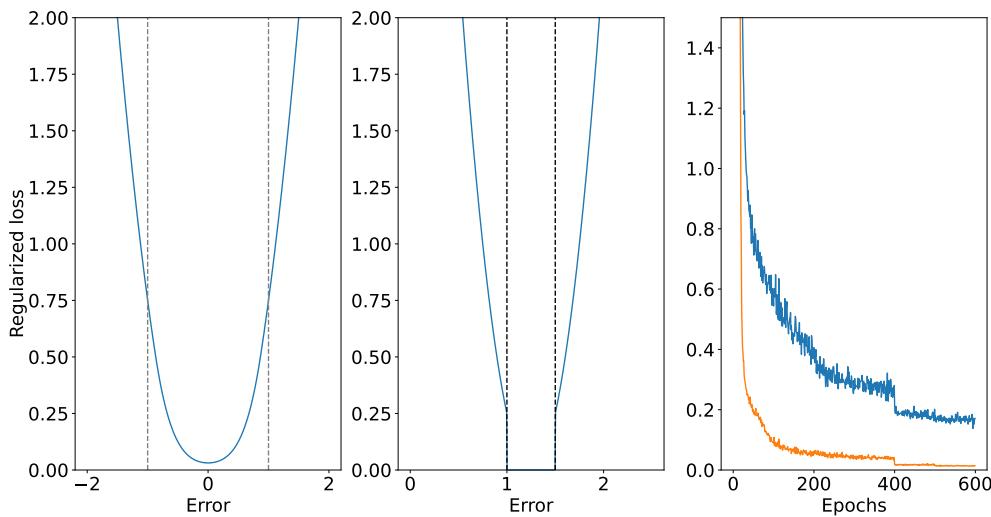


Figure 9. Gas-powered system regularized loss illustration. Left: node balance and Weymouth penalties based on conventional Huber-loss. Middle: Compression factor limit constraint using our Huber-based enhancement. (see Eq. 10). Right: Gas-powered system custom penalty evolution (Blue: Weymouth equality constraint; Orange: compression ratio limit constraint).

In turn, we designed three evaluation scenarios in comparison with the IPOPT framework to validate the performance of regularization functions in data generation. In the first scenario, data remain below the source's maximum capacity. In the second scenario, 50 percent of the samples exceed this capacity, while in the third, about 100 percent of the data surpasses it. Figure 10 shows that even though IPOPT has a lower MAPE, its precision (variance) changes a lot over the iterations. This means that conventional methods for NOPT are not strong against data variability and nonlinear constraints. In contrast, our RPINN achieves acceptable MAPE with low variability across experiments due to its regularized strategy based on ANNs. In fact, both approaches share similar costs and adhere to compression ratio constraints. In the first two cases, traditional solutions to the Weymouth equation work better than ours. But in the third case, our proposal is better because it is more stable and less affected by outliers, thanks to the Huber-based penalty.

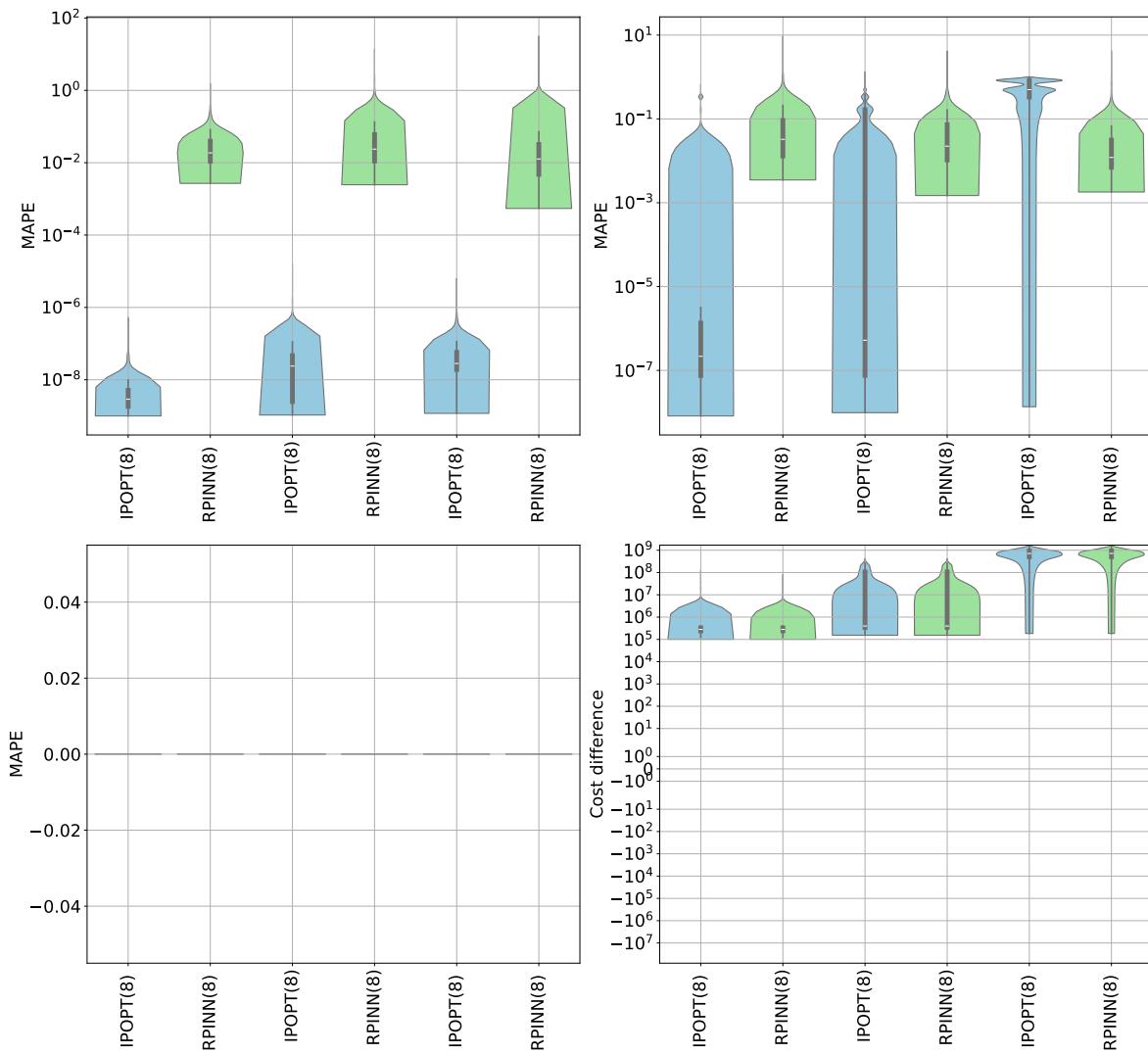


Figure 10. Gas-powered system objective cost and constraint compliance MAPE results. Upper left: node balance. Upper right: Weymouth constraint. Bottom left: compression ratio constraint. Bottom right: cost difference (objective function) between RPINN and IPOPT.

Finally, to see if our RPINN model can handle the limits in Eq. 9 well, we looked at the results of the flow and pressure prediction layers and how they behaved, as shown in Figure 11. The parameters analyzed, including injection and pipe flows as well as pipeline pressures, remain within acceptable limits. This behavior is attributed to custom activation in Eq. 13, which ensures a smooth and steady transition between the established ranges.

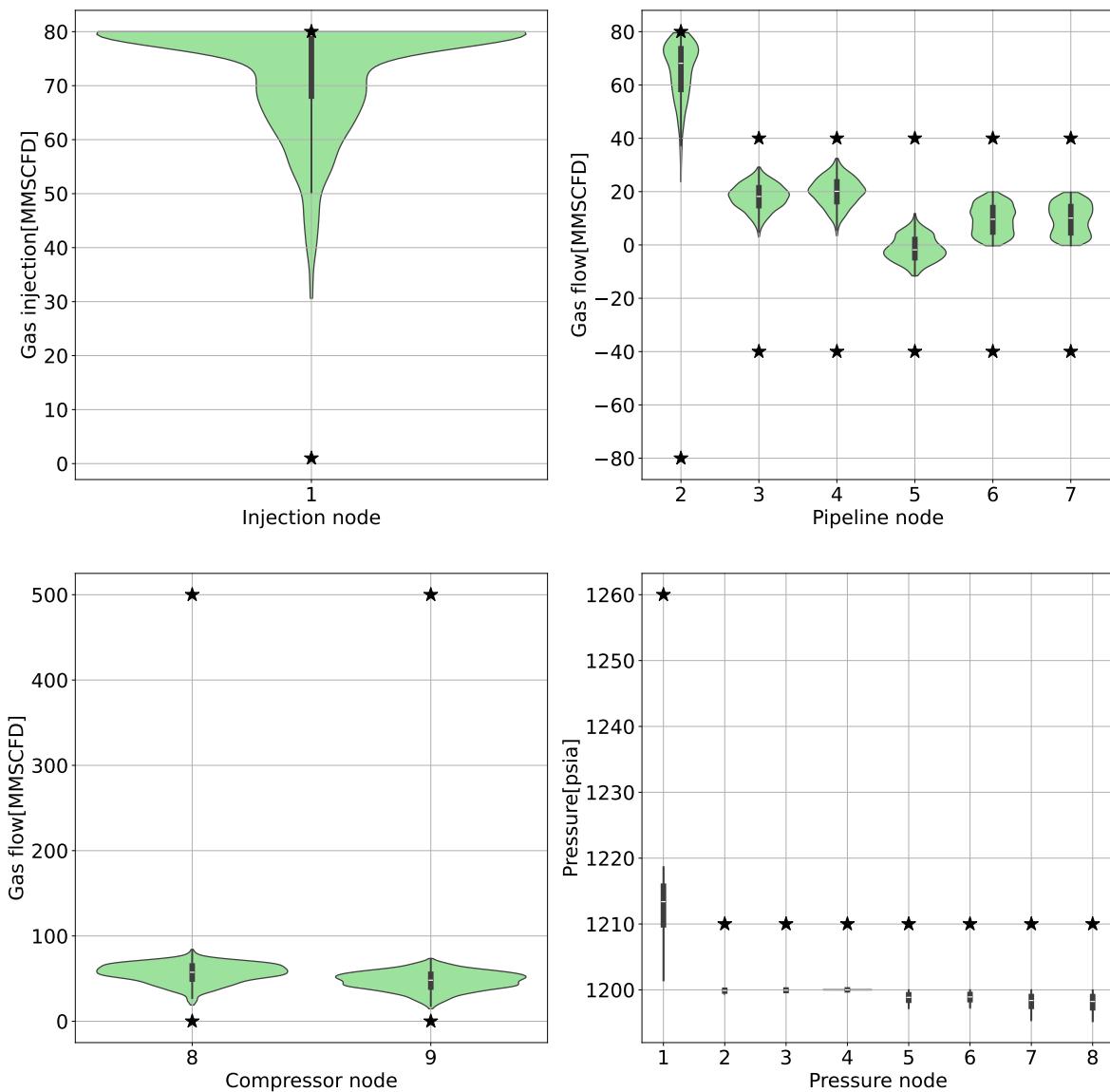


Figure 11. Gas-powered system bound constraint MAPE results. The star symbol on this graph denotes the defined limits for each of the sources, compressors, pipelines, and pressures as well as their behavior. The number on the x-axis indicates the node to which the information belongs. MMSCFD: Million standard cubic feet per day. psia: pounds per square inch absolute.

6.3. Computational Cost Results

Figure 12 shows the training and prediction times needed by the RPINN compared to IPOPT. Our model needs more time to process during the training phase because it has to do both forward and backward passes in each iteration within an ANN-based framework. However, in the prediction phases, our RPINN outperforms IPOPT, resulting in significantly shorter prediction times. This is due to the fact that the model only requires forward passes after training the weights. These tools demonstrate the RPINN's capability to generate fast and accurate predictions for NOPT solutions, not only by reducing processing times but also by narrowing interquartile ranges.

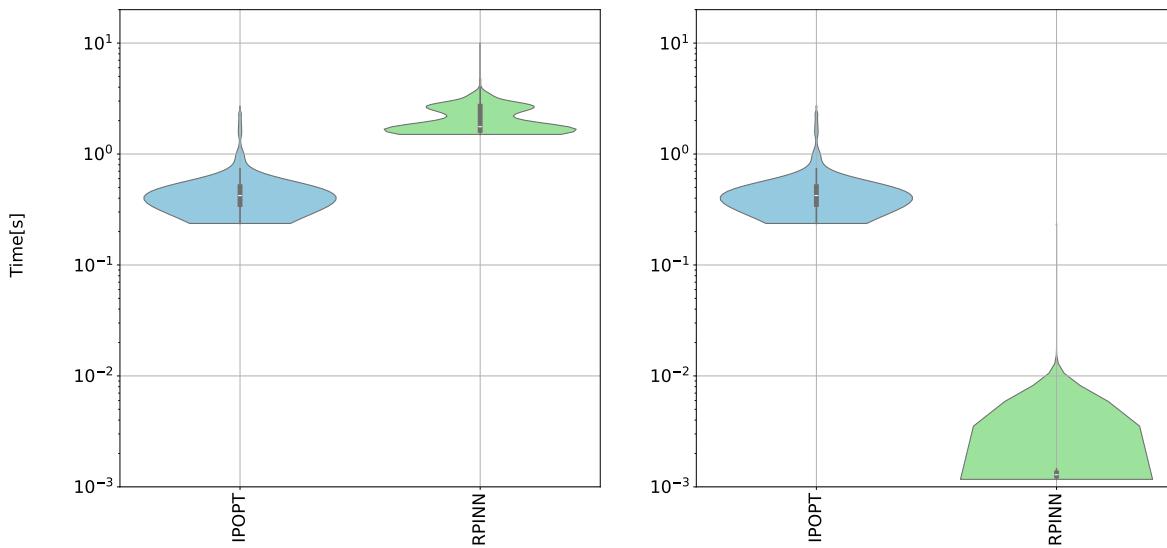


Figure 12. RPINN vs. IPOPT computational cost results. The graph compares solution times for the test data between the classical technique (IPOPT, in blue) and our strategy (RPINN, in green). On the left, the training times are shown, while on the right, the prediction times are displayed.

6.4. Limitations

The RPINN framework, while innovative and effective in addressing many challenges of NOPT, has several limitations that need to be considered. One significant limitation is the complexity involved in defining appropriate loss functions and selecting optimal hyperparameters, which can make the implementation process cumbersome. Additionally, extremely high levels of noise or complex nonlinear constraints can hinder the performance of RPINN, despite its robustness against data variability and noisy inputs. Although AD has improved the model's scalability, it may still face challenges when applied to very large-scale problems due to computational resource limitations.

Furthermore, integrating precise physical principles into the network architecture can be intricate and may not always generalize well across different types of NOPT problems. Current trends in PINNs emphasize improving these models' generalization capabilities and computational efficiency [76]. To better solve the problems of scalability and accuracy, researchers are focusing on hybrid approaches that mix PINNs with other advanced optimization methods, like metaheuristics and gradient-based methods. The latter indicates a growing recognition of the need for more flexible and adaptive frameworks that can handle a broader range of NOPT scenarios.

7. Conclusions

We introduce a novel Regularized Physics-Informed Neural Network (RPINN) framework, named RPINN, presenting a significant advancement in addressing the challenges associated with nonlinear constrained optimization. By integrating custom activation functions and regularization penalties within an ANN architecture, RPINN effectively handles data variability and noisy inputs. Besides, the incorporation of physics principles into the network architecture allows for the computation of optimization variables based on network weights and learned features, leading to competitive performance compared to state-of-the-art solvers. Furthermore, the use of automatic differentiation for training enhances scalability and reduces computation time, making RPINN a robust solution for various NOPT tasks. Experimental results included two scenarios regarding supervised and unsupervised datasets.

The uniform mixture model experiments (supervised constrained NOPT) show that the RPINN is good at dealing with data variability and noisy samples. For noise-free data, both RPINN and the IPOPT solver achieved similar results due to the convex nature of the problem. Still, in scenarios with noisy inputs, RPINN significantly outperformed IPOPT. The RPINN framework, leveraging the

Huber loss function, showed greater robustness against noise by effectively regularizing the network weights. This resulted in more accurate and stable output predictions compared to IPOPT, which relied on an objective function based on the l_2 -norm and was more sensitive to outliers. The RPINN weight distributions were concentrated, which showed that the model could find the main output dynamics even when noise was present, as shown by the lower mean absolute percentage error across all signal-to-noise ratio values.

Then, the results of the gas-powered system (unsupervised constrained optimization) highlight the capability of the RPINN framework to effectively manage complex, nonlinear constraints under varying conditions of gas demand. Compared to the IPOPT framework, the RPINN showed consistent performance with low changes in the mean absolute percentage error. This was especially true when the gas demand was higher than the source's maximum capacity. While IPOPT showed lower MAPE in terms of node balance and Weymouth constraints, its precision fluctuated significantly with data variability. In contrast, RPINN maintained stable performance, ensuring compliance with physical constraints such as the Weymouth equation and compression ratio limits. The custom penalty functions within RPINN facilitated this stability, proving particularly valuable when traditional methods struggled with outliers and extreme values. Overall, RPINN offered a robust, scalable solution with reduced prediction times.

As future work, authors plan to include Bayesian hyperparameter optimization for RPINN fine tuning [77]. We will also look at normalized and information theoretic learning-based loss as ways to deal with noisy inputs and complicated constraints [78,79]. Finally, Bayesian PINN and graph neural networks will be coupled with our RPINN for representation learning enhancement [68,80].

Author Contributions: Conceptualization, D.P.-R., A.A.-M. and G.C.-D.; data curation, D.P.-R.; methodology, D.P.-R., A.A.-M., and G.C.-D.; project administration, A. A.-M.; supervision, A.A.-M. and G. C.-D.; resources, D.P.-R. and A. A.-M. All authors have read and agreed to the published version of the manuscript.

Funding: Under grants provided by the projects: "Desarrollo de una herramienta para la planeación a largo plazo de la operación del sistema de transporte de gas natural en Colombia" (Minciencias-contrato 184-2021) and "Sistema prototípico de visión por computador utilizando aprendizaje profundo como soporte al monitoreo de zonas urbanas desde unidades aéreas no tripuladas - HERMES 55261" (Universidad Nacional de Colombia).

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The publicly available dataset analyzed in this study can be found at <https://github.com/UN-GCPDS/python-gcpds.optimization> (accessed on 1 March 2024).

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

References

1. Branislav, I.; Haifeng, M.; Dijana, M.; others. A survey of gradient methods for solving nonlinear optimization. *Electronic research archive* **2020**, *28*, 1573–1624.
2. Abdulkadirov, R.; Lyakhov, P.; Nagornov, N. Survey of optimization algorithms in modern neural networks. *Mathematics* **2023**, *11*, 2466.
3. Chen, Q.; Zuo, L.; Wu, C.; Bu, Y.; Lu, Y.; Huang, Y.; Chen, F. Short-term supply reliability assessment of a gas pipeline system under demand variations. *Reliability Engineering & System Safety* **2020**, *202*, 107004.
4. Yu, W.; Huang, W.; Wen, Y.; Li, Y.; Liu, H.; Wen, K.; Gong, J.; Lu, Y. An integrated gas supply reliability evaluation method of the large-scale and complex natural gas pipeline network based on demand-side analysis. *Reliability Engineering & System Safety* **2021**, *212*, 107651.
5. Kohjitani, H.; Koda, S.; Himeno, Y.; Makiyama, T.; Yamamoto, Y.; Yoshinaga, D.; Wuriyanghai, Y.; Kashiwa, A.; Toyoda, F.; Zhang, Y.; others. Gradient-based parameter optimization method to determine membrane ionic current composition in human induced pluripotent stem cell-derived cardiomyocytes. *Scientific Reports* **2022**, *12*, 19110.

6. Shcherbakova, G.; Krylov, V.; Qianqi, W.; Rusyn, B.; Sachenko, A.; Bykovyy, P.; Zahorodnia, D.; Kopania, L. Optimization methods on the wavelet transformation base for technical diagnostic information systems. 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). IEEE, 2021, Vol. 2, pp. 767–773.
7. Weiner, A.; Semaan, R. Backpropagation and gradient descent for an optimized dynamic mode decomposition. *arXiv preprint arXiv:2312.12928* **2023**.
8. Han, M.; Du, Z.; Yuen, K.F.; Zhu, H.; Li, Y.; Yuan, Q. Walrus optimizer: A novel nature-inspired metaheuristic algorithm. *Expert Systems with Applications* **2024**, *239*, 122413.
9. Mhanna, S.; Mancarella, P. An exact sequential linear programming algorithm for the optimal power flow problem. *IEEE Transactions on Power Systems* **2021**, *37*, 666–679.
10. Chang, H.; Chen, Q.; Lin, R.; Shi, Y.; Xie, L.; Su, H. Controlling Pressure of Gas Pipeline Network Based on Mixed Proximal Policy Optimization. 2022 China Automation Congress (CAC). IEEE, 2022, pp. 4642–4647.
11. Wang, G.; Zhao, W.; Qiu, R.; Liao, Q.; Lin, Z.; Wang, C.; Zhang, H. Operational optimization of large-scale thermal constrained natural gas pipeline networks: A novel iterative decomposition approach. *Energy* **2023**, *282*, 128856.
12. Montoya, O.; Gil-González, W.; Hernández, J.C.; Giral-Ramírez, D.A.; Medina-Quesada, A. A mixed-integer nonlinear programming model for optimal reconfiguration of DC distribution feeders. *Energies* **2020**, *13*, 4440.
13. Robuschi, N.; Zeile, C.; Sager, S.; Braghin, F. Multiphase mixed-integer nonlinear optimal control of hybrid electric vehicles. *Automatica* **2021**, *123*, 109325.
14. Arya, A.K.; Jain, R.; Yadav, S.; Bisht, S.; Gautam, S. Recent trends in gas pipeline optimization. *Materials Today: Proceedings* **2022**, *57*, 1455–1461.
15. Sadat, S.A.; Sahraei-Ardakani, M. Customized sequential quadratic programming for solving large-scale ac optimal power flow. 2021 North American Power Symposium (NAPS). IEEE, 2021, pp. 1–6.
16. Awwal, A.M.; Kumam, P.; Abubakar, A.B. A modified conjugate gradient method for monotone nonlinear equations with convex constraints. *Applied Numerical Mathematics* **2019**, *145*, 507–520.
17. Gao, H.; Li, Z. A benders decomposition based algorithm for steady-state dispatch problem in an integrated electricity-gas system. *IEEE Transactions on Power Systems* **2021**, *36*, 3817–3820.
18. Wang, Y.; Gao, S.; Zhou, M.; Yu, Y. A multi-layered gravitational search algorithm for function optimization and real-world problems. *IEEE/CAA Journal of Automatica Sinica* **2020**, *8*, 94–109.
19. Pillutla, K.; Roulet, V.; Kakade, S.M.; Harchaoui, Z. Modified Gauss-Newton Algorithms under Noise. 2023 IEEE Statistical Signal Processing Workshop (SSP), 2023, pp. 51–55. <https://doi.org/10.1109/SSP53291.2023.10207977>.
20. Jamii, J.; Trabelsi, M.; Mansouri, M.; Mimouni, M.F.; Shatanawi, W. Non-Linear Programming-Based Energy Management for a Wind Farm Coupled with Pumped Hydro Storage System. *Sustainability* **2022**, *14*. <https://doi.org/10.3390/su141811287>.
21. Baydin, A.G.; Pearlmutter, B.A.; Radul, A.A.; Siskind, J.M. Automatic differentiation in machine learning: A survey. *Journal of machine learning research* **2018**, *18*, 1–43.
22. Pan, X.; Chen, M.; Zhao, T.; Low, S.H. DeepOPF: A Feasibility-Optimized Deep Neural Network Approach for AC Optimal Power Flow Problems. *IEEE Systems Journal* **2023**, *17*, 673–683. <https://doi.org/10.1109/JSYST.2022.3201041>.
23. Nellikkath, R.; Chatzivasileiadis, S. Physics-informed neural networks for ac optimal power flow. *Electric Power Systems Research* **2022**, *212*, 108412.
24. Huang, B.; Wang, J. Applications of Physics-Informed Neural Networks in Power Systems - A Review. *IEEE Transactions on Power Systems* **2023**, *38*, 572–588. <https://doi.org/10.1109/TPWRS.2022.3162473>.
25. Stiasny, J.; Chevalier, S.; Chatzivasileiadis, S. Learning without data: Physics-informed neural networks for fast time-domain simulation. 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 2021, pp. 438–443.
26. Strelow, E.L.; Gerisch, A.; Lang, J.; Pfetsch, M.E. Physics informed neural networks: A case study for gas transport problems. *Journal of Computational Physics* **2023**, *481*, 112041.

27. Applegate, D.; Diaz, M.; Hinder, O.; Lu, H.; Lubin, M.; O' Donoghue, B.; Schudy, W. Practical Large-Scale Linear Programming using Primal-Dual Hybrid Gradient. *Advances in Neural Information Processing Systems*; Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; Vaughan, J.W., Eds. Curran Associates, Inc., 2021, Vol. 34, pp. 20243–20257.

28. Zhao, Z.; Liu, S.; Zhou, M.; Abusorrah, A. Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem. *IEEE/CAA Journal of Automatica Sinica* **2020**, *8*, 1199–1209.

29. Vo, T.Q.T.; Baiou, M.; Nguyen, V.H.; Weng, P. Improving Subtour Elimination Constraint Generation in Branch-and-Cut Algorithms for the TSP with Machine Learning. *Learning and Intelligent Optimization*; Sellmann, M.; Tierney, K., Eds.; Springer International Publishing: Cham, 2023; pp. 537–551.

30. Sun, Y.; Zhang, B.; Ge, L.; Sidorov, D.; Wang, J.; Xu, Z. Day-ahead optimization schedule for gas-electric integrated energy system based on second-order cone programming. *CSEE Journal of Power and Energy Systems* **2020**, *6*, 142–151.

31. Lin, Y.; Zhang, X.; Wang, J.; Shi, D.; Bian, D. Voltage Stability Constrained Optimal Power Flow for Unbalanced Distribution System Based on Semidefinite Programming. *Journal of Modern Power Systems and Clean Energy* **2022**, *10*, 1614–1624.

32. Chowdhury, M.M.U.T.; Kamalasadan, S. A new second-order cone programming model for voltage control of power distribution system with inverter-based distributed generation. *IEEE Transactions on Industry Applications* **2021**, *57*, 6559–6567.

33. Asgharieh Ahari, S.; Kocuk, B. A mixed-integer exponential cone programming formulation for feature subset selection in logistic regression. *EURO Journal on Computational Optimization* **2023**, *11*, 100069. <https://doi.org/https://doi.org/10.1016/j.ejco.2023.100069>.

34. Kumar, J.; Rahaman, O. Lower bound limit analysis using power cone programming for solving stability problems in rock mechanics for generalized Hoek–Brown criterion. *Rock Mechanics and Rock Engineering* **2020**, *53*, 3237–3252.

35. Lin, Y.; Zhang, X.; Wang, J.; Shi, D.; Bian, D. Voltage Stability Constrained Optimal Power Flow for Unbalanced Distribution System Based on Semidefinite Programming. *Journal of Modern Power Systems and Clean Energy* **2022**, *10*, 1614–1624. <https://doi.org/10.35833/MPCE.2021.000220>.

36. Abubakar, A.B.; Kumam, P. A descent Dai-Liao conjugate gradient method for nonlinear equations. *Numerical Algorithms* **2019**, *81*, 197–210.

37. Chen, J.; Wang, L.; Wang, C.; Yao, B.; Tian, Y.; Wu, Y.S. Automatic fracture optimization for shale gas reservoirs based on gradient descent method and reservoir simulation. *Advances in Geo-Energy Research* **2021**, *5*, 191–201.

38. Mahapatra, D.; Rajan, V. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. *International Conference on Machine Learning*, PMLR, 2020, pp. 6597–6607.

39. Karimi, M.; Shahriari, A.; Aghamohammadi, M.; Marzooghi, H.; Terzija, V. Application of Newton-based load flow methods for determining steady-state condition of well and ill-conditioned power systems: A review. *International Journal of Electrical Power & Energy Systems* **2019**, *113*, 298–309.

40. Mannel, F.; Rund, A. A hybrid semismooth quasi-Newton method for nonsmooth optimal control with PDEs. *Optimization and Engineering* **2021**, *22*, 2087–2125.

41. Pinheiro, R.B.; Balbo, A.R.; Cabana, T.G.; Nepomuceno, L. Solving Nonsmooth and Discontinuous Optimal Power Flow problems via interior-point ℓ_p -penalty approach. *Computers & Operations Research* **2022**, *138*, 105607.

42. Delgado, J.A.; Baptista, E.C.; Balbo, A.R.; Soler, E.M.; Silva, D.N.; Martins, A.C.; Nepomuceno, L. A primal-dual penalty-interior-point method for solving the reactive optimal power flow problem with discrete control variables. *International Journal of Electrical Power & Energy Systems* **2022**, *138*, 107917.

43. Liu, B.; Yang, Q.; Zhang, H.; Wu, H. An interior-point solver for AC optimal power flow considering variable impedance-based FACTS devices. *IEEE Access* **2021**, *9*, 154460–154470.

44. Haji, S.H.; Abdulazeez, A.M. Comparison of optimization techniques based on gradient descent algorithm: A review. *PalArch's Journal of Archaeology of Egypt/Egyptology* **2021**, *18*, 2715–2743.

45. Ibrahim, I.A.; Hossain, M.J. Low voltage distribution networks modeling and unbalanced (optimal) power flow: A comprehensive review. *IEEE Access* **2021**, *9*, 143026–143084.

46. Goulart, P.; Chen, Y. Clarabel Documentation. <https://oxfordcontrol.github.io/ClarabelDocs/stable/>, 2024. Último acceso en 2024.

47. Gurobi Optimization. <https://www.gurobi.com/>, 2024. Último acceso en 2024.

48. MOSEK. <https://www.mosek.com/>, 2024. Último acceso en 2024.

49. Xpress Optimization. <https://www.fico.com/en/products/fico-xpress-optimization>, 2024. Último acceso en 2024.

50. O'Donoghue, B. Operator Splitting for a Homogeneous Embedding of the Linear Complementarity Problem. *SIAM Journal on Optimization* **2021**, *31*, 1999–2023.

51. Ipopt Deprecated Features. <https://coin-or.github.io/Ipopt/deprecated.html>, 2024. Último acceso en 2024.

52. Zimmerman, R.D.; Murillo-Sánchez, C.E. *MATPOWER User's Manual*. Zenodo, 2020. <https://doi.org/10.5281/zenodo.4074122>.

53. Wang, H.; Murillo-Sánchez, C.E.; Zimmerman, R.D.; Thomas, R.J. On Computational Issues of Market-Based Optimal Power Flow. *IEEE Transactions on Power Systems* **2007**, *22*, 1185–1193. <https://doi.org/10.1109/TPWRS.2007.901301>.

54. García-Marín, S.; González-Vanegas, W.; Murillo-Sánchez, C. MPNG: A MATPOWER-Based Tool for Optimal Power and Natural Gas Flow Analyses. *IEEE Transactions on Power Systems* **2022**, pp. 1–9. <https://doi.org/10.1109/TPWRS.2022.3195684>.

55. Beal, L.; Hill, D.; Martin, R.; Hedengren, J. GEKKO Optimization Suite. *Processes* **2018**, *6*, 106. <https://doi.org/10.3390/pr6080106>.

56. Mugel, S.; Kuchkovsky, C.; Sanchez, E.; Fernandez-Lorenzo, S.; Luis-Hita, J.; Lizaso, E.; Orus, R. Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks. *Physical Review Research* **2022**, *4*, 013006.

57. Diamond, S.; Boyd, S. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* **2016**, *17*, 1–5.

58. Agrawal, A.; Boyd, S. Disciplined quasiconvex programming. *Optimization Letters* **2020**. To appear.

59. O'Donoghue, B.; Chu, E.; Parikh, N.; Boyd, S. Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding. *Journal of Optimization Theory and Applications* **2016**, *169*, 1042–1068.

60. Pan, X.; Zhao, T.; Chen, M.; Zhang, S. DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow. *IEEE Transactions on Power Systems* **2021**, *36*, 1725–1735. <https://doi.org/10.1109/TPWRS.2020.3026379>.

61. Baker, K. A learning-boosted quasi-newton method for ac optimal power flow. *arXiv preprint arXiv:2007.06074* **2020**.

62. Zhou, M.; Chen, M.; Low, S.H. DeepOPF-FT: One Deep Neural Network for Multiple AC-OPF Problems With Flexible Topology. *IEEE Transactions on Power Systems* **2023**, *38*, 964–967. <https://doi.org/10.1109/TPWRS.2022.3217407>.

63. Liang, H.; Zhao, C. DeepOPF-U: A Unified Deep Neural Network to Solve AC Optimal Power Flow in Multiple Networks, 2023, [\[arXiv:cs.LG/2309.12849\]](https://arxiv.org/abs/2309.12849).

64. Falconer, T.; Mones, L. Leveraging Power Grid Topology in Machine Learning Assisted Optimal Power Flow. *IEEE Transactions on Power Systems* **2023**, *38*, 2234–2246. <https://doi.org/10.1109/TPWRS.2022.3187218>.

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

66. Misyris, G.S.; Stiasny, J.; Chatzivasileiadis, S. Capturing power system dynamics by physics-informed neural networks and optimization. 2021 60th IEEE Conference on Decision and Control (CDC). IEEE, 2021, pp. 4418–4423.

67. Habib, A.; Yildirim, U. Developing a physics-informed and physics-penalized neural network model for preliminary design of multi-stage friction pendulum bearings. *Engineering Applications of Artificial Intelligence* **2022**, *113*, 104953.

68. Yang, L.; Meng, X.; Karniadakis, G.E. B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics* **2021**, *425*, 109913.

69. Schiassi, E.; De Florio, M.; D'Ambrosio, A.; Mortari, D.; Furfaro, R. Physics-informed neural networks and functional interpolation for data-driven parameters discovery of epidemiological compartmental models. *Mathematics* **2021**, *9*, 2069.

70. Raynaud, G.; Houde, S.; Gosselin, F.P. ModalPINN: An extension of physics-informed Neural Networks with enforced truncated Fourier decomposition for periodic flow reconstruction using a limited number of imperfect sensors. *Journal of Computational Physics* **2022**, *464*, 111271.
71. Nellikkath, R.; Chatzivasileiadis, S. Physics-Informed Neural Networks for AC Optimal Power Flow. *Electric Power Systems Research* **2022**, *212*, 108412. <https://doi.org/10.1016/j.epsr.2022.108412>.
72. Murphy, K.P. *Probabilistic machine learning: An introduction*; MIT press, 2022.
73. González-Vanegas, W.; Álvarez Meza, A.; Hernández-Muriel, J.; Orozco-Gutiérrez, Á. AKL-ABC: An Automatic Approximate Bayesian Computation Approach Based on Kernel Learning. *Entropy* **2019**, *21*. <https://doi.org/10.3390/e21100932>.
74. García-Marín, S.; González-Vanegas, W.; Murillo-Sánchez, C. MPNG: MATPOWER-Natural Gas. <https://github.com/MATPOWER/mpng>, 2019. [Online; accessed (fecha de acceso)].
75. Owerko, D.; Gama, F.; Ribeiro, A. Unsupervised optimal power flow using graph neural networks. *arXiv preprint arXiv:2210.09277* **2022**.
76. Mustajab, A.H.; Lyu, H.; Rizvi, Z.; Wuttke, F. Physics-Informed Neural Networks for High-Frequency and Multi-Scale Problems Using Transfer Learning. *Applied Sciences* **2024**, *14*, 3204.
77. Eleftheriadis, P.; Leva, S.; Ogliari, E. Bayesian hyperparameter optimization of stacked bidirectional long short-term memory neural network for the state of charge estimation. *Sustainable Energy, Grids and Networks* **2023**, *36*, 101160.
78. Ma, X.; Huang, H.; Wang, Y.; Romano, S.; Erfani, S.; Bailey, J. Normalized loss functions for deep learning with noisy labels. International conference on machine learning. PMLR, 2020, pp. 6543–6553.
79. Jeon, H.J.; Van Roy, B. An Information-Theoretic Framework for Deep Learning. *Advances in Neural Information Processing Systems* **2022**, *35*, 3279–3291.
80. Thangamuthu, A.; Kumar, G.; Bishnoi, S.; Bhattoo, R.; Krishnan, N.; Ranu, S. Unravelling the performance of physics-informed graph neural networks for dynamical systems. *Advances in Neural Information Processing Systems* **2022**, *35*, 3691–3702.

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