frontiers | Frontiers in Chemical Engineering

TYPE Original Research
PUBLISHED 14 January 2026
DOI 10.3389/fceng.2025.1687048

OPEN ACCESS

EDITED BY
Quinn Reynolds,
Mintek, South Africa

REVIEWED BY
Dongheon Lee,
FAMU-FSU Department of Chemical and Biomedical Engineering, United States
Yongfei Xue,
Central South University of Forestry and Technology, China

*CORRESPONDENCE
Anderson C. Faller,
faller@petrobras.com.br

RECEIVED 16 August 2025
REVISED 13 October 2025
ACCEPTED 18 November 2025
PUBLISHED 14 January 2026

CITATION
Faller AC, Vieira SC and Castro MSd (2026) Steady-state 1D two-phase flow differentiable modeling: learning from field data and inverse problem applications in oil wells.
Front. Chem. Eng. 7:1687048.
doi: 10.3389/fceng.2025.1687048

Steady-state 1D two-phase flow differentiable modeling: learning from field data and inverse problem applications in oil wells

Anderson C. Faller[1,2]*, Saon C. Vieira[1] and Marcelo S. de Castro[2,3]

[1]Petróleo Brasileiro S.A. - PETROBRAS, Santos, Brazil, [2]School of Mechanical Engineering, Universidade Estadual de Campinas - UNICAMP, Campinas, Brazil, [3]Centro de Estudos de Energia e Petróleo - CEPETRO, Universidade Estadual de Campinas - UNICAMP, Campinas, Brazil

Accurately modeling steady-state two-phase flow is critical for the design and operation of systems in the oil and gas industry; however, traditional models often struggle to adapt to specific field conditions. This study introduces a novel, end-to-end differentiable framework that integrates physics-informed neural networks with a Neural Ordinary Differential Equation (Neural ODE) formulation to predict pressure and temperature profiles. By leveraging automatic differentiation, the entire simulation functions as a trainable model, allowing for the simultaneous optimization of data-driven components and the automated tuning of physical parameters directly from field data. Our results demonstrate that this approach achieves superior accuracy in pressure prediction compared to tuned industry-standard correlations. We found that a transfer learning strategy, pretraining on a large experimental dataset to establish a robust physical foundation, followed by fine-tuning on sparse field data, significantly outperforms models trained on field data alone. Furthermore, the differentiable nature of the framework enabled seamless application to inverse problems, demonstrated via Randomized Maximum Likelihood (RML) for uncertainty quantification. These findings illustrate the effectiveness of bridging the domain gap between experimental and real-world conditions, presenting a powerful new paradigm for creating self-calibrating, data-driven simulation tools with significant potential for digital twin applications.

KEYWORDS

two-phase flows, simulation, automatic differentiation, machine learning, physics-informed, neural networks, oil and gas production

1 Introduction

The simultaneous flow of gas and liquid inside a pipeline, known as "two-phase flow," is a fundamental phenomenon across several industries, including oil and gas production, chemical processing, and nuclear power generation (Ishii and Hibiki, 2006). The effective design, optimization, and safe operation of these systems are critically dependent on a thorough understanding of their complex hydrodynamic behavior. Key to this understanding is the characterization of void fraction, pressure, and temperature profiles. The pressure and temperature gradients are influenced by a complex interplay of interfacial and wall friction, gravitational effects, acceleration, fluids properties, pipe characteristics as diameter, roughness, and inclination, and the spatial distribution of the phases—known as the flow pattern (Shoham, 2006). Concurrently, the void fraction (the

cross-sectional area ratio occupied by the gas phase) is a determining factor for the flow pattern, which in turn governs the system's pressure drop and heat transfer characteristics. Accurately predicting these parameters is therefore essential for tasks such as equipment sizing, production optimization, and the prevention of operational hazards. Although many industrial flows are transient, steady-state analysis where flow parameters are constant over time is indispensable for modeling these complex systems.

The development of accurate multiphase flow models has evolved significantly. As classified by Brill and Arirachakaran (1992) and reviewed by Shippen and Bailey (2012), the development can be broadly categorized into three eras. The "Empirical Period" (c. 1950–1975) was characterized by correlations derived from experimental data, with seminal works like Lockhart and Martinelli (1949) providing the initial framework. These models, while foundational, were inherently limited to the conditions under which they were developed. The subsequent "Awakening Years" (c. 1975–1985) marked a shift toward incorporating more fundamental physics and behaviors, with models such as the influential correlation by Beggs and Brill (1973), which first systematically accounted for pipe inclination and flow pattern. This evolution culminated in the "Mechanistic Modeling Era" (post-1985), where models were constructed from first principles using the conservation of mass, momentum, and energy, although experimental data for the closure models base on experimental data are still required. Pioneering models from Taitel and Dukler (1976) for flow pattern prediction and later comprehensive models from research groups at The University of Tulsa and Tel Aviv University provided more robust and generalizable tools by grounding their predictions in the underlying physical phenomena. This research culminated in the unified mechanistic model by Gomez et al. (2000), which integrated and extended previous specialized models for different flow patterns—such as Taitel and Barnea (1990) on slug flow, Alves et al. (1991) on upward annular flow, and Taitel and Dukler (1976) on stratified flow in horizontal and near-horizontal pipes—into a single comprehensive framework applicable to the entire range of pipe inclinations.

In recent years, the proliferation of computational power and data has catalyzed the application of machine learning (ML) to address the persistent challenges in multiphase flow modeling (Zhu et al., 2022; Brunton et al., 2020). AI-based methodologies have been successfully employed to predict critical parameters such as flow patterns, pressure drop, and void fraction with a high degree of accuracy, consistently outperforming traditional empirical correlations when sufficient training data are available (Attia et al., 2015; Alakbari et al., 2025). For instance, in pressure drop prediction, an adaptive neuro-fuzzy inference system (ANFIS) model achieved an average absolute percentage error (AAPE) as low as 0.39% (Attia et al., 2015), while Alakbari et al. (2025) demonstrated a coefficient of determination (R^2) of 0.9645, which proved superior to established correlations.

A diverse array of supervised learning algorithms has been deployed. Artificial neural networks (ANNs) are the most prevalent, valued for their flexibility in modeling non-linear relationships for void fraction prediction (achieving an R^2 of 0.985) (Ruiz-Diaz et al., 2021), bottomhole pressure for oil and gas wells (with errors as low as 2.5%) (Khan et al., 2023), and flow pattern classification (with accuracy as high as 85.97%) (Ezzatabadipour et al., 2017). Support vector machines (SVMs) are also robustly applied to both classification and regression tasks (Attia et al., 2015; Guillen-Rondon et al., 2018). Furthermore, ensemble methods have shown exceptional performance; random forest, for example, was identified as a top-performing algorithm for flow pattern classification, with an F1-score of 0.908 (Alhashem, 2019), while gradient boosting has yielded an R^2 of 0.95 for pressure gradient prediction (Kanin et al., 2019).

The input features for these models typically include superficial velocities, pipe geometry, and fluid properties similar to the above models. However, purely data-driven models often struggle to generalize beyond their training domain and lack physical interpretability. This has led to the emergence of hybrid approaches that integrate physics-based knowledge with data-driven learning, aiming to create models that are both accurate and robust (Raissi et al., 2019). A central enabler of this integration is differentiable modeling, where the entire model is constructed such that its outputs are differentiable with respect to its inputs (Baydin et al., 2018). This property not only facilitates the efficient training of complex neural networks via gradient-based optimization but also establishes novel pathways for solving inverse problems and performing uncertainty quantification. This represents the next frontier in the evolution of multiphase flow simulation.

The physical problem of predicting steady-state pressure and temperature profiles along a pipeline can be described as a system of ordinary differential equations (ODEs) where the independent variable is the spatial coordinate (see Section 2.2). The neural ordinary differential equation (Neural ODE) framework, introduced by Chen et al. (2018), is designed to learn the continuous dynamics of such systems. Therefore, we adopt this paradigm because it provides a mathematically congruent and powerful method for modeling the continuous spatial evolution of the fluid's properties throughout the pipeline in a fully differentiable manner. Lai et al. (2021) proposed the use of physics-informed Neural ODEs for structural identification, employing a similar approach, although within a different application.

Our proposed methodology, which is the subject of a pending patent application (Castro et al., 2025b) and software registration (Castro et al., 2025a), is designed to self-calibrate using field measurements, thus enabling the automated tuning of both physical and data-driven model components. The remainder of this article is organized as follows: Section 2 details the core components of our framework, including the implementation of automatic differentiation, the Neural ODE structure, the specific physics-informed model architecture, and the datasets employed. Section 3 presents the results, evaluating the model's performance, analyzing the effects of different training strategies, and demonstrating its application to Bayesian inference. Finally, Section 4 discusses the key findings, and Section 5 concludes by summarizing contributions and outlining avenues for future research.

2 Methods

2.1 Automatic differentiation

Training complex models, such as proposed in this work, relies on gradient-based optimization methods to minimize a loss

function. Automatic differentiation (AD) is a set of computational techniques for efficiently and accurately calculating the derivatives of functions specified by a computer program. Unlike symbolic differentiation, which can lead to inefficiently large expressions, or numerical differentiation, which is prone to round-off and truncation errors, AD computes exact derivatives by systematically applying the chain rule at the level of an elementary operations computational graph (Baydin et al., 2018).

AD operates in two primary modes: forward and reverse. Reverse mode AD, also known as "backpropagation," is particularly well-suited for machine learning applications. It computes the gradient of a scalar output (such as the loss function $\mathcal{L}$) with respect to a large number of input parameters (e.g., neural network weights and physical parameters) in a single pass through the computational graph.

In this work, the entire model, from the fluid property look-ups and the evaluation of physical equations to the neural network components and the ordinary differential equation (ODE) solver, is implemented within PyTorch (Paszke et al., 2019), a framework that supports AD. This creates an end-to-end differentiable pipeline. Consequently, we can compute the gradients of the final loss function $\mathcal{L}$ with respect to all trainable parameters simultaneously. This includes not only the weights and biases of the neural networks (for estimating pressure gradients, void fraction, and flow pattern) but also physical parameters such as pipe roughness and the heat transfer coefficient. This powerful capability transforms the entire simulation into a trainable model, allowing for the simultaneous optimization of data-driven components and the tuning of physical parameters against measured data. This approach is conceptually related to physics-informed neural networks (PINNs) (Raissi et al., 2019) and libraries like DeepXDE (Lu et al., 2021), which also leverage automatic differentiation to integrate physical laws. However, a key distinction lies in the method of integration. A standard PINN typically embeds physical laws as soft constraints by adding the residuals of the governing differential equations to the loss function. Our framework, in contrast, builds the physical equations directly into the model's architecture. For instance, as shown in Section 2.3, the gravitational component of the pressure drop is explicitly calculated from first principles, and the neural network is tasked only with learning the closure relationship for the frictional component. This hard-coded, physics-structured approach ensures that fundamental principles are satisfied by construction, offering a more constrained and interpretable model for this class of engineering problem.

2.2 Neural ODEs

Traditional deep learning models, such as residual networks (ResNets; He et al., 2016), can be interpreted as discrete approximations of a continuous transformation, being similar to a first order approximation scheme, for example. Each layer applies a discrete step transformation $h_{t+1} = h_t + f(h_t, \beta_t)$ to the hidden state h. Neural ordinary differential equations (Neural ODEs) generalize this concept by parameterizing the continuous dynamics of a system's hidden state with a neural network (Chen et al., 2018). Instead of a sequence of discrete layers, a Neural ODE defines the derivative of the state with respect to time (or another continuous variable) using a neural network (Equation 1).

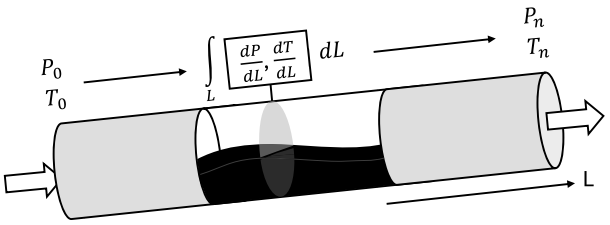

FIGURE 1
Illustration of a two-phase flow through a pipe section in a stratified flow pattern.

$$\frac{dh(t)}{dt} = f(h(t), t, \beta). \tag{1}$$

The final state is then computed by integrating this derivative function using a numerical ODE solver while allowing gradient propagation through the solution. This framework offers several advantages, including memory-efficient training through the adjoint sensitivity method.

The determination of thermohydraulic profiles for one-dimensional, two-phase flow within a wellbore is fundamentally rooted in the conservation laws of mass, momentum, and energy. Mathematically, these laws are formulated as a system of coupled partial differential equations (PDEs). When conditions such as pressure and temperature are specified at two distinct boundary locations along the pipeline, such as at the bottomhole and wellhead, the problem is posed as a classical boundary value problem (BVP) with a Dirichlet boundary condition (Ishii and Hibiki, 2006). For steady-state analysis, a classical and robust numerical technique for solving such BVPs is the shooting method (Kong et al., 2021). This method effectively transforms the BVP into an initial value problem (IVP) by assuming initial conditions at one boundary and integrating forward or backward to satisfy the conditions at the second boundary iteratively. Consequently, the steady-state system can be modeled as a system of ODEs where the spatial coordinate along the pipe serves as the independent variable (Shoham, 2006).

Within steady-state multiphase flow applications, the pressure and temperature profiles in a pipeline are traditionally calculated using a specialized method called a "marching algorithm" (Shoham, 2006), where the pipeline is discretized in small segments, and the fluid properties are calculated iteratively based on the average pressure and temperature in each segment until the pressure drop converges. The calculation then "marches" to the next segment, until the last one, similar to a predictor corrector method. In this context, the Neural ODE paradigm is exceptionally well-suited. The evolution of the fluid's pressure P and temperature T along the length of a pipeline L, as noted above, can be framed as an initial value problem (IVP) if their values are known at one end, the flow rate is set beforehand (not as a result of an equilibrium), the fluid properties are also known, and it is intended to calculate the pressure and temperature profiles and their values at the other end. Defining the pipeline length L as the integration variable (Figure 1), we can model the system's state vector $[P, T]^T$ as the solution to an ODE, where the derivative

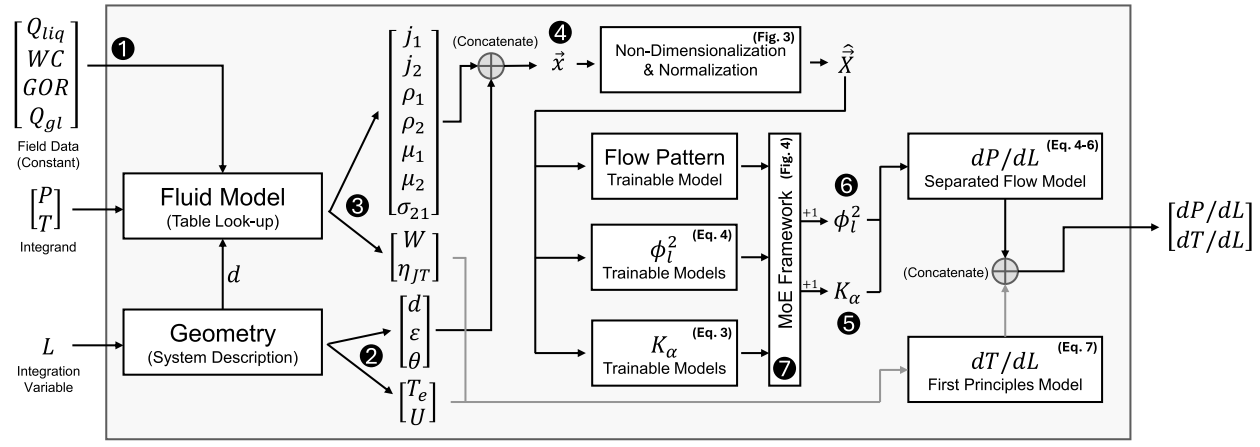

FIGURE 2
Model architecture with highlighted calculation steps.

function f is learned from data; however, it has a physical structure. This allows the model to continuously and accurately predict the pressure and temperature profiles at any point along the pipe, not just at discrete locations, providing a powerful and physically intuitive foundation for our differentiable modeling approach.

It is critical to note that, when applying the Neural ODE framework, the integration variable in our formulation is the pipeline length L, representing a spatial dynamic, not time t; our model is exclusively for steady-state conditions where temporal dynamics are, by definition, absent.

2.3 Model development

Following the Neural ODE paradigm, the problem is mathematically formulated by modeling the evolution of the fluid's pressure and temperature, represented by the state vector $[P, T]^T$, along the pipeline length L. The core of this approach is to learn function f, parameterized by a set of weights β. This function effectively represents the pressure and temperature gradients, leading to the following system of ODEs (Equation 2).

$$\frac{d}{dL}\begin{bmatrix} P \\ T \end{bmatrix} = f\left(\begin{bmatrix} P \\ T \end{bmatrix}, L, \beta\right). \tag{2}$$

The system's geometry and environmental conditions vary along the flow. Therefore, the function is variant with respect to the integration variable. Considering this compact formulation for f, which is better represented in Figure 2, β is the set of the model parameters—the fluid look-up table and the system description—used in the transformation of the input data. Based on flow parameters—Q_{liq}^{SC} (liquid flow rate at standard conditions), WC (water cut), GOR (gas–oil ratio), and Q_{gl} (gas-lift flow rate at standard conditions)—and on geometrical and environmental parameters—d (diameter), θ (pipe inclination), ε (roughness), T_E (ambient temperature), and U (overall heat transfer coefficient)—the *in situ* fluid properties are calculated through a routine that performs bi-linear interpolation on PVT property tables. Such calculated properties are j_1 and j_2 (liquid and gas superficial velocities), ρ_1 and ρ_2 (liquid and gas densities), μ_1 and μ_2 (liquid and gas dynamic viscosities), σ_{21} (surface tension between liquid and gas), C_P (specific heat of the mixture), η_{JT} (Joule–Thomson coefficient of the mixture), and W (total mass flow rate). These parameters are also used as inputs for traditional calculation models in the literature (Shippen and Bailey, 2012), including both empirical and mechanistic approaches. In our framework, they serve as the dimensional inputs for the model.

Although it is possible to use models like multi-layer perceptrons (MLPs) (Werbos, 1974) directly for pressure drop calculation, we introduced a physical structure to the models to increase their generalization capacity. The model implemented calculates the void fraction α using correction factor K_α based on the no-slip void fraction α_{ns} (Equation 3). Chisholm (1983) and Armand (1946) developed correlations for K_α (Pietrzak and Płaczek, 2019). In the model implementation, α must be limited to the interval [0,1].

$$\alpha = K_\alpha \alpha_{ns} = K_\alpha \frac{j_2}{J}, \tag{3}$$

where J is the mixture velocity (sum of gas and liquid superficial velocities), and K_α is obtained from a neural network. Note that, considering a homogeneous model, without slippage, $\alpha = \alpha_{ns}$ and $K_\alpha = 1$. Therefore, it is expected that the estimated values for K_α are approximately 1. In order to ensure that the model produces physically plausible outputs, considering a neural network NN_{K_α} that outputs values approximately 0 when initialized, we defined K_α such that $K_\alpha = 1 + \mathrm{NN}_{K_\alpha}(\hat{\vec{X}})$, where $\hat{\vec{X}}$ is the input features vector described in Section 2.4. This physical structure acts as a regularization technique in the overall model, outputting physically plausible values even after initialization. Other approaches for a physics-oriented α-model were tested, such as a drift-flux model $(\alpha = j_2/(C_0 J + V_{2j}))$ with neural network representations for C_0 and V_{2j}, also with good results. However, the training process was found to be less stable while increasing the model's complexity. Therefore, we adopted the K_α factor approach, as it yielded the best metrics.

For the calculation of the total pressure drop dP/dL, we introduced a physical structure in our model, considering a gravitational term, dependent on the fluid's density and pipe inclination, and the Lockhart–Martinelli separated flow model with the empirical two-phase multiplier ϕ_l^2 (Lockhart and Martinelli, 1949), relating the two-phase flow friction pressure drop to the liquid-only pressure drop (Equation 4) (Shoham, 2006):

$$\frac{dP}{dL} = -\rho_m g \sin\theta - \phi_l^2 \frac{2\rho_1 f_1 j_1 |j_1|}{d}, \tag{4}$$

where g is the gravity, the mixture's density is given by $\rho_m = \rho_1(1-\alpha) + \rho_2\alpha$, and the two-phase multiplier is obtained by a neural network $\mathrm{NN}_{\phi_l^2}(\vec{\tilde{X}})$. Similarly to K_α, ϕ_l^2 benefits the model's stability in the training process when it is equal to 1 after initialization; thus $\phi_l^2 = 1 + \mathrm{NN}_{\phi_l^2}(\vec{\tilde{X}})$. The Fanning friction factor f_1 is obtained with a correlation for single-phase flow. In this implementation, we chose the explicit formula of Swamee and Jain (1976) for the friction factor (Equation 5) with the Reynolds number given by Equation 6. This is because Colebrook–White (Colebrook, 1939) had an implicit formulation which would require additional implementation to allow automatic differentiation, and the high order exponents of the formula of Churchill (1977) have shown numerical instabilities in the backward calculations.

$$f_1 = \frac{0.0625}{\left[\log\left(\frac{\varepsilon}{3.7d} + \frac{5.74}{\mathrm{Re}_1^{0.9}}\right)\right]^2}, \tag{5}$$

$$\mathrm{Re}_1 = \frac{\rho_1 |j_1| d}{\mu_1}. \tag{6}$$

Finally, the thermal calculation derives directly from the energy balance and has an explicit model (Shoham, 2006):

$$\frac{dT}{dL} = \frac{U\pi d}{WC_P}(T_E - T) + \frac{1}{C_P}\left(\eta_{JT}^{\star} C_P \frac{dP}{dL} - g\sin\theta - v\frac{dv}{dL}\right), \tag{7}$$

where the acceleration term $v\frac{dv}{dL}$ is negligible in steady-state flow compared to the other terms, so we disregarded this term in our calculations, thus also avoiding the need for implicit calculations. Additionally, we introduced a linear transformation on the Joule–Thomson coefficient to account for uncertainties in the fluid properties (Equation 8), where a_{JT} and b_{JT} are trainable scalars defined for each well.

$$\eta_{JT}^{\star} = a_{JT} + b_{JT} \times \eta_{JT}. \tag{8}$$

The model also relies on a mixture of experts (MoE) framework with the ability to learn models dependent on the flow pattern (Section 2.5). In addition to the trainable parameters of the K_α and ϕ_l^2 models, the automatic differentiation framework also allows the training of physical parameters, such as the overall heat transfer coefficient U, roughness ε, or coefficients a_{JT} and b_{JT}, which can serve as global adjustment parameters for the models. More broadly, all parameters are trainable and adjustable, making the proposed architecture a powerful method for parameter determination.

The implementation also allows the pretraining of specific parts of the model, such as the α and dP/dL models with experimental bench data. Pretraining was performed with a dataset of experimental data from TUFFP (Tulsa University Fluid Flow Projects), detailed further in Section 2.7.

All the model's implementations and modules allow for gradient propagation in addition to batch inference and training, as well as the use of a GPU for acceleration.

2.4 Learnable dimensionless numbers

The Buckingham Pi theorem is a robust technique for extracting insights and finding symmetries in physical systems when governing equations are absent. It provides a method for finding a set of dimensionless groups (pi-groups) that span the solution space of a physical problem given a set of measurement variables and parameters (Fox et al., 1985). The theorem is based on the idea that physical laws do not depend on the units of measurement. Therefore, any function that expresses a physical law must have generalized homogeneity.

BuckiNet is a deep learning algorithm proposed by Bakarji et al. (2022) that incorporates the Buckingham Pi theorem as a constraint. It aims to automatically discover the dimensionless groups that best collapse measurement data to a lower-dimensional space according to an optimal fit.

The input dimensional parameters $\vec{x}$ are transformed using a logarithmic function. To avoid non-positive values, we take their absolute values and add a small constant ε_d. The weights of the first layer (Φ_p) act as the exponents to form the dimensionless input groups ($\vec{\pi}$) through an exponential activation function at the output of this layer, resulting in the expression for $\vec{\pi}$ in Equation 9.

$$\vec{\pi} = \exp\left(log\left(|\vec{x}| + \varepsilon_d\right)^T \Phi_p\right)^T. \tag{9}$$

To explicitly ensure that the groups discovered are dimensionless, a null space loss $\mathcal{L}_{null}$ can be added to the loss function. This loss penalizes deviations from the condition $D_p\Phi_p = 0$, where D_p is the units matrix (Equation 10) corresponding to the input parameters $\vec{x}$.

$$D_p = \begin{array}{c} \\ \textbf{mass} \\ \textbf{length} \\ \textbf{time} \end{array} \begin{array}{c} \begin{array}{cccccccccc} j_1 & j_2 & \rho_1 & \rho_2 & \mu_1 & \mu_2 & \sigma_{21} & d & \varepsilon & \theta \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & -3 & -3 & -1 & -1 & 0 & 1 & 1 & 0 \\ -1 & -1 & 0 & 0 & -1 & -1 & -2 & 0 & 0 & 0 \end{bmatrix} \end{array} \tag{10}$$

A ℓ_1 regularization term is added to enforce sparsity (desired in dimensionless numbers) and a ℓ_2 regularization term to encourage smaller powers. The training losses related to the BuckiNet are (Equation 11):

$$\mathcal{L}_{BuckiNet} = \lambda_0\mathcal{L}_{null} + \lambda_1\ell_1 + \lambda_2\ell_2 = \lambda_0\|D_p\Phi_p\|_2 + \lambda_1\|\Phi_p\|_1 + \lambda_2\|\Phi_p\|_2. \tag{11}$$

The dimensional $\vec{x}$ and dimensionless $\vec{\pi}$ inputs are then concatenated (Figure 3) to compose the input features of the model. Neural networks are more stable and train faster when the inputs are normalized. However, some of the input features (the dimensionless ones) are learned during training, changing their distribution along the training process (internal covariate shift), requiring the adoption of a trainable normalizer, such that they cannot be normalized beforehand based on a fixed distribution. Some trainable normalizers were tested—batch normalization (Ioffe and Szegedy, 2015) and layer normalization (Ba et al., 2016)—and an

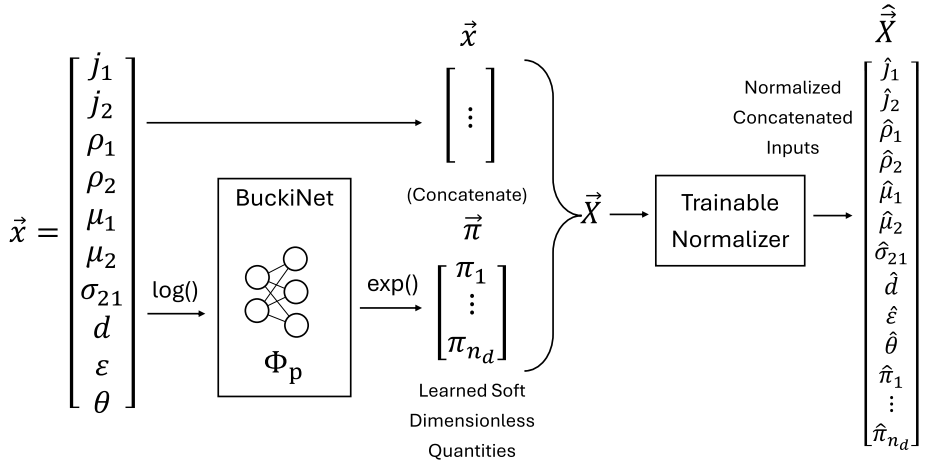

FIGURE 3
Dimensional and learned dimensionless features combined using a trainable normalizer to generate the model's inputs.

alternative to normalization—dynamic hyperbolic tangent (Zhu et al., 2025). Ultimately, after the hyperparameter optimization process was performed, batch normalization (*BatchNorm*) consistently yielded the best metrics.

Batch normalization (*BatchNorm*) is a technique that stabilizes and accelerates neural network training by standardizing the inputs for each layer on a per-mini-batch basis. During training, it calculates the mean and variance of the activations within a mini-batch, normalizes them to have a mean of zero and unit variance, and then applies a learnable scale γ and shift β to restore the network's expressive power. For inference, when mini-batches are not present, it uses a running average of the mean and variance collected during the training phase to perform the same normalization, ensuring consistent output.

Therefore, the input features of the neural networks are the normalized concatenated inputs (Equation 12):

$$\hat{\vec{X}} = \mathrm{BN}_{\gamma,\beta}\left(\left[\vec{x}^T, \vec{\pi}^T\right]^T\right). \tag{12}$$

2.5 Mixture of experts

To improve the model's physical interpretability, we designed it to first predict the flow pattern, since the governing physics are pattern-dependent. This classification is then used to select specialized sub-models for predicting dP/dL and α. This approach also introduces sparsity to the model's representation. While mixture of experts (MoE) is a well-established concept (Nowlan and Hinton, 1991), its application to multiphase flow calculations is novel, exploiting the flow patterns classification as a key to expert selection, unlike the traditional approach where such classification and expert selection is arbitrary. The key idea is to train multiple experts specialized in different tasks (in this case, different flow patterns), then to weight them in the prediction, ultimately increasing generalization. The MoE paradigm is used in language tasks (Shazeer et al., 2017; Fedus et al., 2022).

Depending on the characteristics of the available dataset and how the flow pattern is labeled, a classifier with n_p classes can be designed to represent different flow patterns or groups of them. The output of the classifier is a vector $[p_1, p_2, \ldots, p_{n_p}]$ such that $\sum p_i = 1$. Instead of switching between models in the transitions, we apply a dot product between the classifier output and a vector containing the predictions of all models (Figure 4). In this manner, using the "soft labels" instead of "hard switches," the flow pattern transitions are smooth and the overall model becomes more stable and easily trainable.

For the two-phase multiplier used in Equation 4, the model must output $\phi_l^2 - 1$ (Equation 13):

$$\phi_l^2 - 1 = \sum_{i=1}^{n_p} p_i \left(\phi_{li}^2 - 1\right). \tag{13}$$

In addition, $K_\alpha - 1$ to estimate the void fraction in Equation 3 is obtained similarly (Equation 14):

$$K_\alpha - 1 = \sum_{i=1}^{n_p} p_i \left(K_{\alpha i} - 1\right). \tag{14}$$

This architecture combines a classification model with specialized sub-models, enabling a flexible, accurate, and physics-informed representation of multiphase flow across various patterns. This approach improves generalization and adaptability in a wide range of operating conditions.

In our work, we narrowed the entire range of possible flow patterns into four classes ($n_p = 4$), for the entire range of pipe inclinations (Table 1).

2.6 Experimental dataset

We used an experimental database from TUFFP (Tulsa University Fluid Flows Project) containing the results of several two-phase flow studies performed since 1966 comprising a wide range of pipe

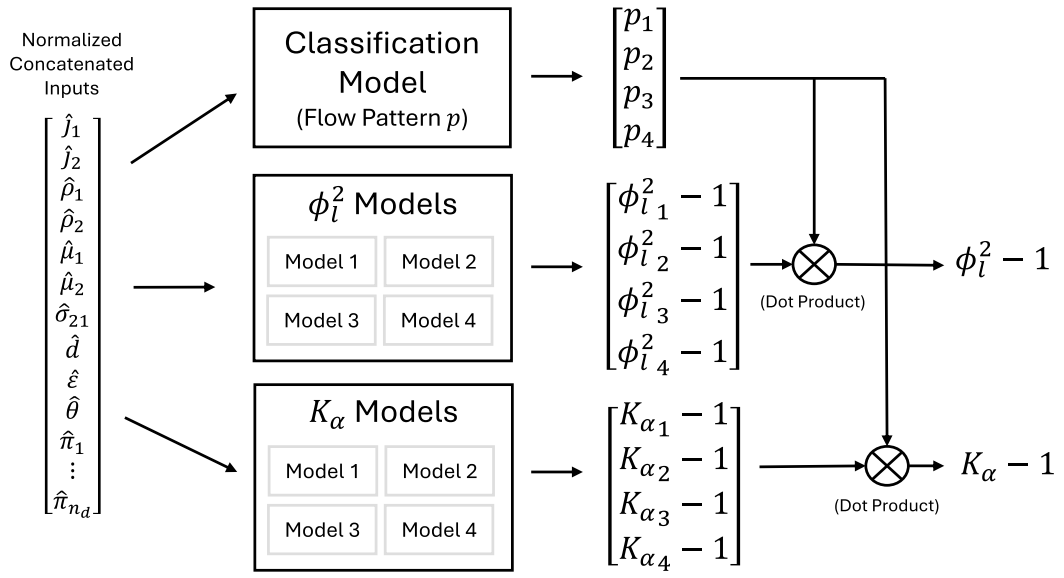

FIGURE 4
Mixture of experts architecture, based on the flow pattern classification model when $n_p = 4$.

TABLE 1 Flow pattern classes ($n_p = 4$).

Class name	Abbreviation	Represented flow pattern
Bubbles	B	Bubbles, dispersed bubbles
Intermittent	I	Slug flow, elongated bubbles, churn
Annular	A	Annular
Stratified	S	Stratified smooth, stratified wavy

TABLE 2 Model input and output physical quantities.

Symbol	Physical quantity	Unit
Inputs		
j_1	Liquid superficial velocity	m/s
j_2	Gas superficial velocity	m/s
ρ_1	Liquid density	kg/m^3
ρ_2	Gas density	kg/m^3
μ_1	Liquid dynamic viscosity	Pa · s
μ_2	Gas dynamic viscosity	Pa · s
σ_{21}	Gas–liquid surface tension	N/m
d	Pipe inner diameter	m
ε	Pipe absolute roughness	m
θ	Pipe inclination angle	rad
Outputs		
α	Void fraction	(–)
dP/dL	Total pressure gradient	Pa/m
FP	Flow pattern class (Table 1)	(–)

inclinations, diameters, fluid velocities, and fluid characteristics. A subset of this dataset was used in the methodology proposed in Pereyra et al. (2012). The dataset used is composed of the data many studies (Abduvayt et al., 2003; Akpan, 1980; Al-Ruhaimani et al., 2017; Almabrok, 2013; Alsaadi, 2013; Andritsos, 1986; Ansari and Azadi, 2016; Barnea et al., 1982, 1985, Beggs, 1972, Chen et al., 1997; Brito, 2012; Brito et al., 2014; Caetano, 1985; Cheremisinoff, 1977; Crowley et al., 1986; Eaton, 1966; Fan, 2005; Felizola, 1992; Gawas, 2005; Gokcal, 2005, 2008, Guner, 2012, Hanafizadeh et al., 2011; Johnson, 2009; Karami, 2015; Khaledi et al., 2014; Kokal, 1987; Kokal and Stanislav, 1989a; Kokal and Stanislav, 1989b; Kouba, 1986; Kristiansen, 2004; Magrini, 2009; Marcano, 1996; Meng, 1999; Minami, 1983; Mukherjee, 1979; Ohnuki and Akimoto, 2000; Roumazeilles, 1994; Schmidt, 1976; Schmidt, 1977; Shmueli et al., 2015; Usui and Sato, 1989; Viana, 2017; Vongvuthipornchai, 1982; Vuong, 2016; Yamaguchi and Yamazaki, 1984; Yang, 1996; Yuan, 2011; Zheng, 1989). The physical quantities used as inputs and outputs in our study are shown in Table 2.

After data processing, removal of entries with missing data, uniformization of the flow patterns to fit in the given definition, and removing a few studies that were not relevant to our field application in oil and gas field data (e.g. microchannel experiments), we were left with 16055 data points, which were split as follows: 60% for model training, 20% for model validation, and 20% for testing (metrics calculation). These splits were stratified based on the flow pattern and inclination range distributions.

2.7 Field dataset

The field dataset is a collection of production tests measurements performed while the well is aligned through a test

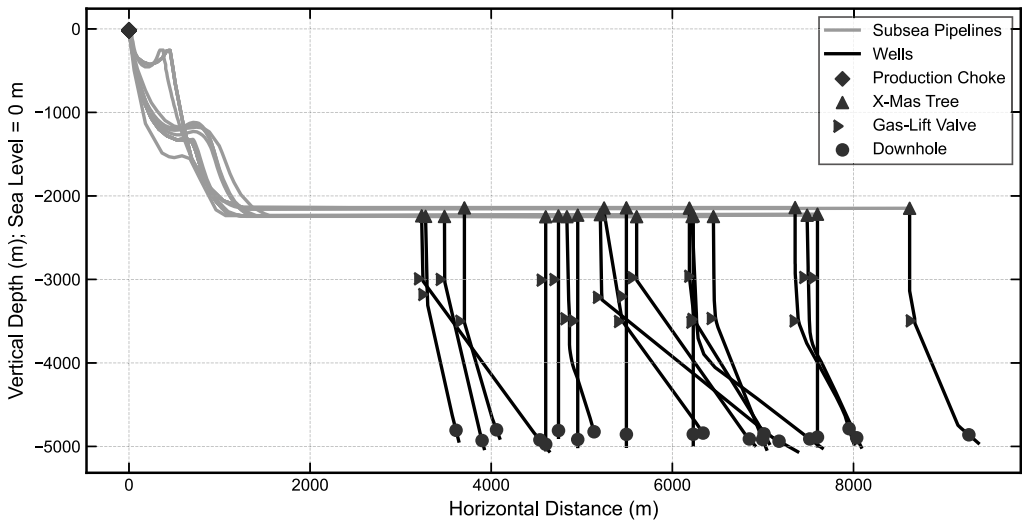

FIGURE 5
Sub-sea wells and pipelines comprising the field dataset geometric data.

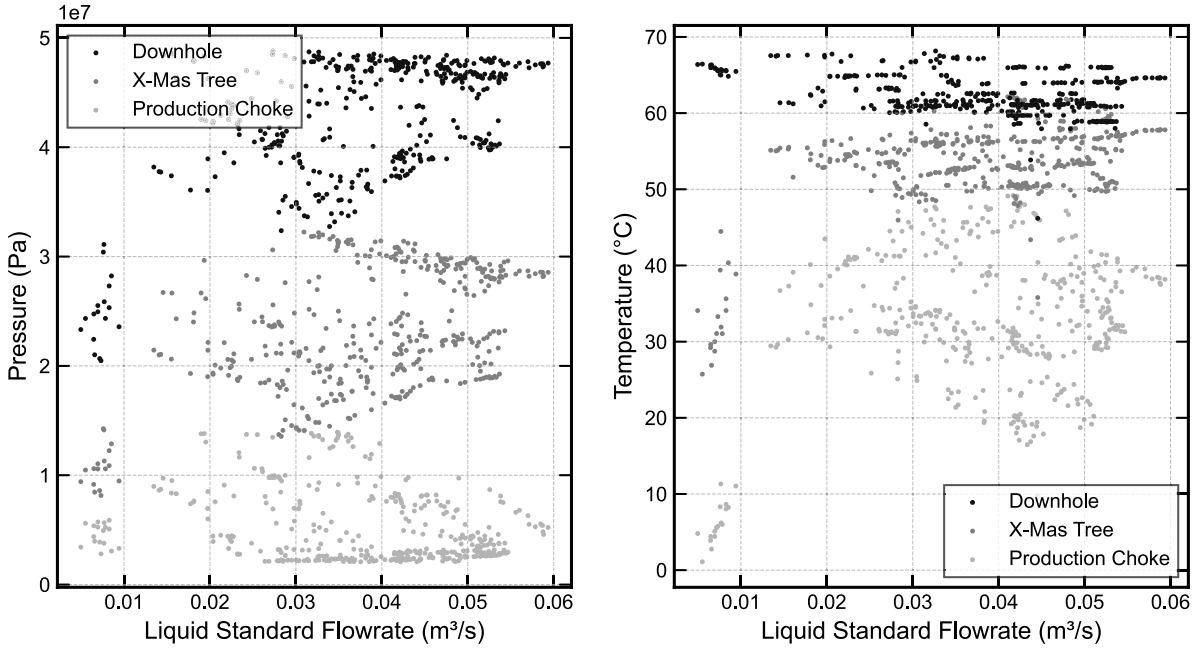

FIGURE 6
Visualization of the field dataset flow rates, pressures, and temperatures.

separator. The multiphase flow rates are assessed from 19 wells connected to three different floating production units in ultra-deep waters. Their geometries and notable points (pressure and temperature sensors and gas-lift valves) are shown in Figure 5. To perform pressure and temperature profiles calculations, we also characterize its diameter, roughness, ambient temperature, and heat transfer coefficient for each pipe section.

The dataset comprises a wide range of liquid (oil + water) flow rates, pressures, and temperatures in each sensor (downhole, Xmas tree, and upstream production choke) shown in Figure 6.

We considered 402 production tests over 4 years which were also split as follows: 60% for model training, 20% for model validation, and 20% for testing (metrics calculation). The splits were stratified by well, so that each set had the same distribution of wells. Additionally, as there is an intermediate sensor (the Xmas tree) along the flow, each production test was used as two distinct data points, considering the flow between the downhole gauge and the Xmas tree (well) and between the Xmas tree and the production choke (pipeline). The relevant measured data contained in each production test is described in Table 3.

TABLE 3 Relevant data from production tests.

Symbol	Description	Unit
Q_{liq}	Liquid standard flow rate	m^3/s
WC	Water cut	(–)
GOR	Gas–oil ratio	(–)
Q_{gl}	Gas-lift standard flow rate[a]	m^3/s
P_{wf}	Downhole flowing pressure	Pa
T_{wf}	Downhole flowing temperature	°C
P_{wh}	Wellhead (Xmas tree) flowing pressure	Pa
T_{wh}	Wellhead (Xmas tree) flowing temperature	°C
P_{ck}	Production choke upstream flowing pressure	Pa
T_{ck}	Production choke upstream flowing temperature	°C

[a]Some wells are naturally flowing (Q_{gl} = 0 m3/s).

It is important to emphasize that the calculations can be performed in two different ways: given the upstream pressures and temperatures, we can calculate the downstream pressures and temperatures, or *vice versa*, as the numerical integration can be done both ways. In this study, we considered the following: given P_{ck} and T_{ck}, calculate P_{wh} and T_{wh}; given P_{wh} and T_{wh}, calculate P_{wf} and T_{wf}.

Finally, each well has an associated fluid description in the form of a tabular TAB file in the same format adopted by commercial simulators such as OLGA, LedaFlow, and Pipesim. These tables are used by the model to calculate the fluids' velocities and properties along the flow as a function of the pressure and temperature in each point. The use of black-oil correlations would also be a possibility, as long as their implementations are fully differentiable, and further adding the capability of adjusting correlation coefficients with field data. The dataset is found attached to this paper.

2.8 Training pipeline

To systematically identify the optimal model configuration and training strategy, a series of experiments was conducted. The investigation involved comparing two primary architectures for the physics-informed closure relations: a standard single multi-layer perceptron (MLP) and a more complex MoE model. For each architecture, we further assessed the influence of feature engineering by evaluating three input configurations: using exclusively dimensional inputs, using only learnable dimensionless numbers, and using a concatenation of both. A central objective was determining the efficacy of transfer learning; thus, we contrasted a training recipe involving pretraining on an experimental dataset against a direct approach using only field data. The performance of each combination was evaluated via the mean absolute error (MAE) for pressure and temperature predictions after each sequential step initialization, pretraining, training, and a final parameter tuning phase and benchmarked against a tuned Beggs and Brill correlation to provide an industry-relevant comparison baseline. A description of each step of the full experiment designed to evaluate the proposed methodology is presented in the following subsections.

2.8.1 Model initialization

The model can be initialized with hyperparameters that determine the specifics of its architecture, such as the α model, dP/dL model, MLP network shapes (number of layers and widths), number of dimensionless numbers, normalization layers, and other relevant parameters. Key aspects include options for enabling an MoE architecture and the choice of how dimensionless numbers are used (concatenated with dimensional inputs, only dimensionless, or no dimensionless numbers).

A hyperparameter optimization process led to the modeling described in Section 2.3, which also providing a range of best performing MLP shapes and number of dimensionless numbers ($n_d > 200$ for the experimental dataset). When comparing the use of MoE to the single MLP for ϕ_l^2 and K_α, the width of the layers was adjusted such that the total number of parameters is, for fair comparison, approximately the same. The number of parameters, as a function of the number of dimensional inputs n_D, number of dimensionless features n_d, number of hidden layers n_h, hidden layer width w_h, number of outputs n_o, and number of experts n_e, is calculated as in Equation 15.

$$\#\,\text{Params} = \left[\underbrace{(n_D + n_d + 1)w_h}_{\text{Input Layer}} + \underbrace{w_h\,(w_h + 1)\,(n_h - 1)}_{\text{Hidden Layers}} + \underbrace{(w_h + 1)n_o}_{\text{Output Layer}}\right] \times \underbrace{n_e}_{\text{Number of Experts}}. \tag{15}$$

The best model configurations found and their calculated number of trainable parameters are shown in Table 4. Note that the number of parameters for the K_α and ϕ_l^2 models are similar for the MoE and single options when using $w_h\,(\text{Single}) = 2 \times w_h\,(\text{MoE})$.

2.8.2 Pretraining with experimental dataset

This section details the pretraining phase, the first major step in our experimental pipeline, designed to establish the model's fundamental predictive capabilities on a comprehensive experimental dataset. The objective here is twofold: first, to confirm that in the proposed physics-informed architectures, both the single MLP and the MoE variants, can effectively learn the underlying closure relations for pressure gradient, void fraction, and flow pattern classification from controlled, data-rich conditions. Second, this phase generates the pretrained models that serve as the starting point for the subsequent transfer learning experiments on field data.

This step is optional because, due to the model's differentiable implementation, it is able to learn directly from field data and does not require pretraining. However, if so set, the initialized model undergoes pretraining using the experimental dataset described in Section 2.6. This step involves training the FP, ϕ_l^2, and K_α models simultaneously, besides the dimensionless matrix Φ_p.

The loss function is composed of cross-entropy loss $\mathcal{L}_{FP}$ (Equation 16) for the flow pattern prediction, ℓ_1 losses $\mathcal{L}_{dP/dL}$ (Equation 17) and $\mathcal{L}_\alpha$ (Equation 18) for the dP/dL and α predictions (weighted by w_c: the inverse of the respective flow

TABLE 4 Hyperparameter configuration for MoE vs. single models.

					MoE				Single			
Model	Dimensionless	n_d	n_D	n_o	n_e	n_h	w_h	# Params	n_e	n_h	w_h	# Params
K_α	No	0	10	1	4	4	600	4.36×10^6	1	4	1,200	4.34×10^6
ϕ_l^2		0	10	1	4	3	2,400	4.62×10^7	1	3	4,800	4.61×10^7
FP		0	10	4	1	3	1700	5.81×10^6	-	-	-	-
K_α	Only	334	0	1	4	4	600	5.13×10^6	1	4	1,200	4.73×10^6
ϕ_l^2		334	0	1	4	3	2,400	4.93×10^7	1	3	4,800	4.77×10^7
FP		334	0	4	1	3	1700	6.36×10^6	-	-	-	-
K_α	Concatenated	334	10	1	4	4	600	5.16×10^6	1	4	1,200	4.74×10^6
ϕ_l^2		334	10	1	4	3	2,400	4.94×10^7	1	3	4,800	4.78×10^7
FP		334	10	4	1	3	1700	6.38×10^6	-	-	-	-

TABLE 5 Loss function weighting hyperparameters.

Training step	λ_{FP}	$\lambda_{dP/dL}$	λ_α	λ_0	λ_1	λ_2	γ
Pretraining	1.05	8.0×10^{-5}	0.1	2.0	1.4	0.3	-
Training (pretrained model)	-	-	-	-	-	-	18000
Training (not pretrained)	-	-	-	5.5×10^5	950	18500	18000
Model parameters tuning	-	-	-	-	-	-	18000
Factors tuning (Beggs and Brill)	-	-	-	-	-	-	10000

pattern frequency), and the BuckiNet losses (null space, ℓ_1, and ℓ_2) described in Section 2.4. n_b is the batch size.

$$\mathcal{L}_{FP} = -\frac{1}{n_b}\sum_{i=1}^{n_b}\sum_{c=1}^{n_p} y_{ic} log\left(p_{ic}\right) + \left(1 - y_{ic}\right) log\left(1 - p_{ic}\right) \tag{16}$$

$$\mathcal{L}_{dP/dL} = \frac{1}{\sum_{i=0}^{n_b} w_i}\sum_{i=1}^{n_b}\left|\frac{dP}{dL_i} - \frac{dP^*}{dL_i}\right| w_i \tag{17}$$

$$\mathcal{L}_{\alpha} = \frac{1}{\sum_{i=0}^{n_b} w_i}\sum_{i=1}^{n_b}\left|\alpha_i - \alpha_i^*\right| w_i. \tag{18}$$

Thus, the training loss function becomes

$$\mathcal{L} = \lambda_{FP}\mathcal{L}_{FP} + \lambda_{dP/dL}\mathcal{L}_{dP/dL} + \lambda_\alpha \mathcal{L}_\alpha + \mathcal{L}_{BuckiNet}, \tag{19}$$

where the weighting factors of the loss function in Equation 19 (λ_{FP}, $\lambda_{dP/dL}$, and λ_α) and the BuckiNet regularization (λ_0, λ_1, and λ_2) were determined through hyperparameter optimization. We used the Optuna framework (Akiba et al., 2019), which employs Bayesian optimization to efficiently search the parameter space, followed by minor manual adjustments to ensure training stability. The optimal weights used in each training step are compiled in Table 5. The model's quality is assessed using the micro F_1 score for the flow pattern classification and the MAE for the dP/dL and α predictions.

Other hyperparameters found via Optuna, such as learning rate lr, batch size n_b, and the choice of optimizer, are listed in Table 6. The AdamW (Loshchilov and Hutter, 2019) optimizer was found suitable for our application due to its weight regularization capability, given the model's complexity. Note that in the parameters tuning step, different learning rates were adopted for different groups of parameters (the learning rate for the heat transfer coefficient U, lr_U, is much larger), mainly because physical parameters are not normalized and their absolute values differ by orders of magnitude.

Due to the complex landscape of the loss function, the training process can frequently get stuck in local minima, requiring perturbations and some noise during training. Finding suitable learning rates and loss ponderation factors can be challenging. The use of a learning rate scheduler that employs cosine annealing with warm restarts (Loshchilov and Hutter, 2016), in composition with a decaying exponential (Equation 20), stabilizes the loss function in the last epochs, brings optimal fitting for all the training objectives.

$$lr^\star = \frac{lr}{8.8} + \frac{\left(lr - \frac{lr}{8.8}\right)}{2}\left(1 + \cos\left(\frac{epoch \bmod 133}{133}\pi\right)\right) 0.997^{epoch}. \tag{20}$$

Training the FP, ϕ_l^2, and K_α models sequentially was also considered. However, the dimensionless matrix Φ_p can only be learned if all objectives are pursued simultaneously because any variation in the dimensionless numbers' definitions will degrade previously trained models.

2.8.3 Training with field data

The goal is to represent the observed field data, matching the model's predictions with the measured pressures and temperatures.

TABLE 6 General training hyperparameters, n_{fb} is the full batch size.

Training step	lr	n_b	Optimizer	Scheduler
Pretraining (single)	1.74×10^{-4}	3211 ($n_{fb}/3$)	AdamW	Equation 20
Pretraining (MoE)	8.71×10^{-4}	3211 ($n_{fb}/3$)	AdamW	Equation 20
Training	5.73×10^{-4}	161 ($n_{fb}/3$)	Adagrad	Equation 22
Model parameters tuning	1.0×10^{-7} ($lr_U = 0.1$)	161 ($n_{fb}/3$)	Adagrad	Equation 22
Factors tuning (Beggs and Brill)	5.0×10^{-2}	121 ($n_{fb}/4$)	Adagrad	Equation 22

As described in Section 2.7, each production test represents two data points in the dataset: besides the other parameters from the test, given P_{ck} and T_{ck}, we predict P_{wh} and T_{wh}; given P_{wh} and T_{wh}, we predict P_{wf} and T_{wf}.

The training loss is the sum of the pressure and temperature ℓ_1 loss, weighted by a factor γ. If the model was pretrained, it is only possible to update the weights of the ϕ_l^2 and α models, as the flow patterns are not available in the field data.

$$\mathcal{L} = \sum_{i=1}^{n_b} \frac{|P_i - P_i^*|}{n_b} + \gamma \sum_{i=1}^{n_b} \frac{|T_i - T_i^*|}{n_b} + \mathcal{L}_{BuckiNet}. \tag{21}$$

When training directly on field data, without pretraining, the flow pattern classification model (FP) and the dimensionless matrix Φ_p should also be trained, besides ϕ_l^2 and K_α models. In this case, the BuckiNet regularization losses $\mathcal{L}_{BuckiNet}$ must also be added to the training loss.

For the training process, a cosine annealing with warm restarts (Loshchilov and Hutter, 2016) learning rate (Equation 22) yields the best results. Large batches ($n_b \geqslant n_{fb}/3$, or even full-batch) have also been shown to result in more stable training curves.

$$lr^\star = \frac{lr}{140} + \frac{\left(lr - \frac{lr}{140}\right)}{2}\left(1 + \cos\left(\frac{epoch \bmod 21}{21}\pi\right)\right). \tag{22}$$

Due to the complexity of the model, which encompasses a set of different sub-models (Φ_p, K_α, ϕ_l^2, FP, and eventually trainable physical parameters; Section 2.8.4), in addition to the sparsity of the available data for training, Adagrad (Duchi et al., 2011) was the most suitable choice of optimizer.

The model quality is assessed using the pressure and temperature MAEs.

2.8.4 Parameter tuning

The previous step (model training) is responsible for adjusting the model's weights, assuming the correctness of the given data, such as pipe diameter, roughness, heat transfer coefficient, and ambient temperature. Our approach, on the other hand, which relies on automatic differentiation, allows optimization of an objective function adjusting any parameter from the model, including the inputs, leading to a parameter estimation framework.

For example: a pipeline's global heat transfer coefficient can be tuned to fit all the historic values of measured temperatures; the equivalent roughness from the flexible pipelines can be tuned to match the measured pressures, where one can share this parameter across all the pipelines from each supplier that have the same diameter, or across all the well tubings; a pipe's equivalent diameter can be calculated as a function of time to estimate the possibility of wax deposition; if properly modeled, coefficients from a black-oil correlation can be tuned to account for the field data, given uncertainties in the fluid composition.

In our study, we tuned several key parameters to account for uncertainties in the physical system and fluid representation. These parameters included the roughness ε of similar pipe groups (well tubing, flexible pipeline, and steel catenary riser), the global heat transfer coefficient U for different pipe types (well tubing, flexible flowline, flexible riser, steel catenary riser), and a trainable linear transformation (Equation 8) for the fluid's Joule–Thomson coefficient η_{JT} for each well. The Joule–Thomson coefficient was specifically chosen for tuning due to the strong coupling observed between pressure and temperature in our field data. This effect dominates the temperature gradient dT/dL in Equation 7, especially in well-insulated pipelines. Since our fluid model used pre-simulated tables with no trainable parameters, the linear transformation was introduced as a necessary mechanism to adjust this critical fluid characteristic.

The loss function is the same as that used in the model training (Equation 21).

2.8.5 Baseline comparison

For comparison, we established a baseline model which implements the Beggs and Brill (1973) correlation, widely used in the oil and gas industry. To simulate the engineering process of tuning the model, we added correction factors for the calculated friction dP/dL, the calculated dT/dL, and the calculated liquid holdup ($H_L = 1 - \alpha$), similar to the factors allowed by commercial simulators such as Pipesim. The latter correction factor is shared between pipes with similar inclinations (horizontal, ascending, or descending). All these factors were limited to the interval $[0.85, 1.15]$, which is a reasonable constraint when adjusting models based on historical data.

2.9 Bayesian inference - RML

The differentiable nature of our model's implementation enables the application of the randomized maximum likelihood (RML) algorithm (Oliver et al., 1996; Bardsley et al., 2014) to efficiently sample the posterior probability distribution of uncertain parameters, given a set of observations and the simulated values.

Consider the model g having a set of uncertain parameters $\vec{m}$ (with dimension N_m) with a known prior normal distribution with mean $\vec{m}_{pr}$ and covariance matrix $\mathbf{C}_m$. The measured data $\vec{d}_{obs}$ (with

dimension N_d) also follow a normal distribution. $\mathbf{C}_e$ is the sum of the covariance matrix of the measurement errors and the covariance matrix of the model errors, considering an imperfect model. From Bayes' rule (Tarantola, 2005) we develop Equation 23:

$$p\left(\vec{m}|\vec{d}_{obs}\right) = \frac{p\left(\vec{d}_{obs}|\vec{m}\right)p\left(\vec{m}\right)}{p\left(\vec{d}_{obs}\right)} = \frac{p\left(\vec{d}_{obs}|\vec{m}\right)p\left(\vec{m}\right)}{\int p\left(\vec{d}_{obs}|\vec{m}\right)p\left(\vec{m}\right)dm}. \tag{23}$$

For a specific realization of $\vec{d}_{obs}$, the likelihood of $\vec{m}$ is $\mathcal{L}\left(\vec{m}\right) = \mathcal{L}\left(\vec{m}|\vec{d}_{obs}\right) \equiv p\left(\vec{d}_{obs}|\vec{m}\right)$, which is modeled as normally distributed:

$$\mathcal{L}\left(\vec{m}\right) = \mathcal{N}\left(\vec{d}_{obs}, \mathbf{C}_e\right) = \frac{1}{\left[\left(2\pi\right)^{N_d}\det\left(\mathbf{C}_e\right)\right]^{1/2}}\exp\left\{-\frac{1}{2}\left(\vec{d}_{obs} - g\left(\vec{m}\right)\right)^T\mathbf{C}_e^{-1}\left(\vec{d}_{obs} - g\left(\vec{m}\right)\right)\right\}, \tag{24}$$

and the prior probability distribution $p\left(m\right)$ is also modeled as a Gaussian:

$$\begin{aligned} p\left(\vec{m}\right) &= \mathcal{N}\left(\vec{m}_{pr}, \mathbf{C}_m\right) \\ &= \frac{1}{\left[\left(2\pi\right)^{N_m}\det\left(\mathbf{C}_m\right)\right]^{1/2}}\exp\left\{-\frac{1}{2}\left(\vec{m} - \vec{m}_{pr}\right)^T\mathbf{C}_m^{-1}\left(\vec{m} - \vec{m}_{pr}\right)\right\} \end{aligned} \tag{25}$$

The posterior distribution of the model parameters m, given observation d_{obs}, is as follows:

$$p\left(\vec{m}|d_{obs}\right) = \frac{\mathcal{L}\left(\vec{m}|\vec{d}_{obs}\right)p\left(\vec{m}\right)}{\int\mathcal{L}\left(\vec{m}|\vec{d}_{obs}\right)p\left(\vec{m}\right)dm} = k \times \mathcal{L}\left(\vec{m}\right)p\left(\vec{m}\right), \tag{26}$$

where k is the marginalization factor. Therefore, substituting Equations 24–26, considering $\mathbf{C}_e$ and $\mathbf{C}_m$ are constant matrices, we have the following Equations 27, 28:

$$\begin{aligned} p\left(\vec{m}|\vec{d}_{obs}\right) &= k' \times \exp\left\{-\frac{1}{2}\left(\vec{d}_{obs} - g\left(\vec{m}\right)\right)^T\mathbf{C}_e^{-1}\left(\vec{d}_{obs} - g\left(\vec{m}\right)\right)\right. \\ &\quad \left.-\frac{1}{2}\left(\vec{m} - \vec{m}_{pr}\right)^T\mathbf{C}_m^{-1}\left(\vec{m} - \vec{m}_{pr}\right)\right\} \end{aligned} \tag{27}$$

$$= k' \times exp\left\{-\mathcal{O}\left(m\right)\right\}. \tag{28}$$

Finally, the *maximum a posteriori* (MAP) estimate is defined as the model parameters $\vec{m}$ that maximize $p\left(\vec{m}|\vec{d}_{obs}\right)$, which is equivalent to minimizing the objective function $\mathcal{O}\left(\vec{m}\right)$, as the marginalization factor k' and other constants can be disregarded.

The RML method is characterized by approximating the posterior distribution by drawing samples from the prior and finding m that minimizes $\mathcal{O}\left(\vec{m}\right)$. One sample $\vec{m}_{c,j}$ of the posterior distribution can be obtained by the following algorithm.

1. Sample $\vec{m}_j$ from the prior $\mathcal{N}\left(\vec{m}_{pr}, \mathbf{C}_m\right)$;
2. Sample $\vec{d}_{obs,j}$ from $\mathcal{N}\left(\vec{d}_{obs}, \mathbf{C}_e\right)$;
3. Compute $\vec{m}_{c,j} = \arg\min_m\mathcal{O}_r\left(\vec{m}\right)$.

where the objective function is

$$\begin{aligned} \mathcal{O}_r\left(\vec{m}\right) &= \frac{1}{2}\left(\vec{d}_{obs,j} - g\left(\vec{m}\right)\right)^T\mathbf{C}_e^{-1}\left(\vec{d}_{obs,j} - g\left(\vec{m}\right)\right) \\ &\quad + \frac{1}{2}\left(\vec{m} - \vec{m}_j\right)^T\mathbf{C}_m^{-1}\left(\vec{m} - \vec{m}_j\right), \end{aligned} \tag{29}$$

which can be noted as two separate objectives (Equation 30):

$$\mathcal{O}_r\left(\vec{m}\right) = \mathcal{O}_{r,d}\left(\vec{m}\right) + \mathcal{O}_{r,m}\left(\vec{m}\right), \tag{30}$$

where $\mathcal{O}_{r,d}\left(\vec{m}\right)$ is the data mismatch (the error between the calculated and observed data), and $\mathcal{O}_{r,m}\left(\vec{m}\right)$ is the model mismatch (the error between the model's parameters and their prior distribution). The model mismatch has a regularization role in the objective function of not allowing the model to overfit the observed data.

The samples do not need to be retrieved sequentially, allowing computation of the entire posterior distribution in parallel—another feature enabled by our modeling. Minimizing $\mathcal{O}_r$ using a built-in solver, therefore, approximates the posterior distribution for $\vec{m}$.

The choice of which model parameters to allocate under $\vec{m}$ and $\vec{d}_{obs}$ depends on the specific application. An example will be given in Section 3.6. The uncertain quantities can be either some model parameters such as roughness or some fluid characteristics, or even a missing sensor reading (e.g. faulty x-tree pressure and temperature transmitters).

3 Results

This section sequentially presents the experimental results designed to validate and analyze the proposed differentiable framework. We first conduct a comprehensive comparison of different model architectures (single MLP vs. mixture of experts (MoE) for the multipliers ϕ_l^2 and K_α; or MoE calculating directly the frictional pressure loss and α), feature engineering approaches (dimensional vs. learnable dimensionless inputs), and training strategies (with and without pretraining on experimental data) to identify the most effective configuration for field applications. Following this, we detail the model's baseline performance on the experimental dataset and isolate the specific contribution of the learnable dimensionless numbers. The central results then focus on the model's accuracy on the field dataset, critically examining the impact of pretraining and benchmarking against an industry-standard correlation. This analysis reveals a significant domain gap between experimental and field conditions. Finally, we demonstrate the framework's broader utility by showcasing its capabilities for automated physical parameter tuning and for uncertainty quantification via randomized maximum likelihood (RML), illustrating its potential for self-calibrating simulations parameter estimation.

In Subsection 3.1, we compare all the training strategies and expose and explain all the metrics obtained in the process. In the subsequent subsections, we visualize the training results and explain them qualitatively.

3.1 Training strategy comparison

Along the training pipeline—initialization, pretraining on experimental data and training on field data (with further parameter tuning on field data)—we evaluated the metrics for both experimental (dP/dL and α MAE and R^2 scores; FP F_1 score) and field (P and T MAE) datasets. The goal is to analyze the effect of each architecture on the metrics, as well as the models' behaviors when sequentially trained with distinct objectives.

TABLE 7 Comparison of model configurations and predicted pressure gradient MAE and R^2 score.

Model configuration			Predicted dP/dL MAE (Pa/m)/R^2		
K_α and ϕ_l^2	Dimensionless	Pretrain	Initialized	Pretrained	Trained
Single MLP	No	Yes	1,117.4/−0.124	266.6/0.737	1,209.0/0.609
Single MLP	No	No	1,117.4/−0.124	-	1875.6/0.164
Single MLP	Only	Yes	1,117.5/−0.124	161.3/0.988	1,261.2/0.655
Single MLP	Only	No	1,117.5/−0.124	-	4,697.8/0.095
Single MLP	Concatenated	Yes	1,117.3/−0.124	159.6/0.987[a]	1,387.3/0.461
Single MLP	Concatenated	No	1,117.3/−0.124	-	4,770.2/0.091
MoE	No	Yes	1,117.4/−0.124	284.4/0.603	928.5/0.501
MoE	No	No	1,117.4/−0.124	-	1,312.3/0.276
MoE	Only	Yes	1,117.5/−0.125	168.0/0.986	744.1/0.789
MoE	Only	No	1,117.5/−0.125	-	4,350.1/0.110
MoE	Concatenated	Yes	1,117.4/−0.124	170.8/0.980	593.9/0.792
MoE	Concatenated	No	1,117.4/−0.124	-	4,882.6/0.098
MoE - no physics	Concatenated	Yes	1881.0/−519.020	139.5/0.990[b]	548.3/0.890[c]
MoE - no physics	Concatenated	No	1881.0/−519.020	-	4,330.3/0.203
Beggs and Brill	-	-	1,514.8/0.572	-	-

[a]Best metrics after pretraining with experimental data, among models without physics.
[b]Best metrics after pretraining with experimental data.
[c]Best metrics after training with field data.

We initialized and trained models following the architecture proposed in Sections 2.3, 2.5: we implemented trainable neural networks as the multipliers ϕ_l^2 and K_α, adopting the MoE framework with $n_p = 4$ to account for each flow pattern category (MoE models in the following tables) but also with the option of $n_p = 1$, making no distinction between flow patterns (single MLP models). To measure the effect of incorporating physics knowledge in the form of the multipliers, we also tested models employing neural networks directly as predictors for α and the frictional dP/dL (without multipliers) but still calculating the gravitational portion based on ρ_m (first term of Equation 4). We then referred these models as "No Physics" in the following tables, still using the MoE framework. Lastly, we compared the Beggs and Brill baseline results for comparison.

Table 7 compiles the results for the experimental dataset (test set) dP/dL predictions, based on the combined MAE and R^2 metrics.

The models that employ multipliers K_α and ϕ_l^2 resulted in a smaller absolute error upon initialization. This is credited as a benefit of incorporating the physics knowledge, as their predictions correspond to a liquid-only frictional calculation with a homogeneous model for the gravitational portion because ϕ_l^2 and K_α are close to 1 when the model is initialized. On the other hand, Beggs and Brill yields the best coefficient of determination R^2 compared to the other initialized models, demonstrating its superior ability to explain the variance in dP/dL compared to the other initialized models, which is expected for a previously crafted correlation. The models without physics (without multipliers) output random values after initialization, explaining the worse metrics.

After pretraining, the use of dimensionless numbers (only or concatenated) systematically improved both MAE and R^2 metrics. Single MLPs for K_α and ϕ_l^2 ($n_p = 1$) showed slightly better (yet similar) performance in this pretraining step then the MoE ($n_p = 4$) counterparts, but the latter retained the best metrics after the training step. However, the model without the multipliers (no physics) had the best pretraining performance (MAE = 138.5, $R^2 = 0.890$). We will highlight below how this model overspecializes in the dataset upon which it is trained.

After training with field data (last column on Table 7), a degradation on the experimental dP/dL metrics for all pretrained models is expected, as the experimental data are not used in the training step. This is confirmed in our results. All the models trained directly on field data (without pretraining) result in high dP/dL errors. However, it is noticeable that all models yielded positive R^2 scores for the experimental dP/dL data, meaning that the incorporated physical structure resulted in more physical model outputs.

The experimental α metrics (MAE and R^2) are shown in Table 8. The conclusions are analogous to the dP/dL analysis. The models that employ the physical multipliers K_α and ϕ_l^2 initialize better than those that do not, while Beggs and Brill outperforms them due to the knowledge crafted into the correlation. Pretraining achieved good results in both MAE and R^2 metrics, and also evident is the benefit of the use of dimensionless numbers. All models with (only and concatenated) dimensionless numbers showed similar performance in pretraining and some degradation after training with field data. The best results are bolded in the table.

TABLE 8 Comparison of model configurations and predicted α MAE and R^2 score.

Model configuration			Predicted α MAE/R^2		
K_α and ϕ_l^2	Dimensionless	Pretrain	Initialized	Pretrained	Trained
Single MLP	No	Yes	0.135/0.579	0.051/0.888	0.139/0.573
Single MLP	No	No	0.135/0.579	-	0.104/0.643
Single MLP	Only	Yes	0.135/0.579	0.041/0.929	0.229/0.397
Single MLP	Only	No	0.135/0.579	-	1.105/−0.356
Single MLP	Concatenated	Yes	0.135/0.579	0.042/0.924	0.192/0.458
Single MLP	Concatenated	No	0.135/0.579	-	1.006/−0.312
MoE	No	Yes	0.135/0.579	0.051/0.897	0.114/0.697
MoE	No	No	0.135/0.579	-	0.145/0.324
MoE	Only	Yes	0.135/0.579	0.036/0.933	0.127/0.727
MoE	Only	No	0.135/0.579	-	0.615/−0.551
MoE	Concatenated	Yes	0.135/0.579	**0.038/0.937**	**0.068/0.884**
MoE	Concatenated	No	0.135/0.579	-	0.747/−0.428
MoE -no physics	Concatenated	Yes	0.340/−128.8	0.082/0.760	0.177/0.613
MoE - no physics	Concatenated	No	0.340/−128.8	-	0.656/-1.4×10^9
Beggs and Brill	-	-	0.081/0.798	-	-

Bold indicates pretraining and training metrics from best performing model.

TABLE 9 Comparison of model configurations and predicted flow pattern F_1 score.

Model configuration			Micro F_1 score		
K_α and ϕ_l^2	Dimensionless	Pretrain	Initialized	Pretrained	Trained
MoE	No	Yes	0.093	0.905	-
MoE	No	No	0.093	-	0.093
MoE	Only	Yes	0.105	0.926	-
MoE	Only	No	0.105	-	0.177
MoE	Concatenated	Yes	0.187	**0.927**	-
MoE	Concatenated	No	0.187	-	0.177
MoE - no physics	Concatenated	Yes	0.187	0.920	-
MoE - no physics	Concatenated	No	0.187	-	0.093
Barnea	-	-	0.699	-	-
Random classifier	-	-	0.250	-	-

Bold indicates best metrics.

The other objective of pretraining is the flow pattern classification only when using a MoE framework ($n_p = 4$). The metrics were compared with the Barnea (1987) unified model baseline ($F_1 = 0.699$) and a hypothetical random classifier, picking each flow pattern with 25% probability ($F_1 = 0.250$). The metrics are compiled in Table 9. All MoE configurations achieved similar results upon pretraining, but the use of concatenated dimensionless features and physical multipliers K_α and ϕ_l^2 yielded the best metric ($F_1 = 0.927$). It is important to note that even if the multipliers are not used in the flow pattern estimation, the training of all the dP/dL, α, and FP objectives are carried simultaneously, affecting the results. For example, an error in a training dP/dL sample propagates its gradient to the FP classifier weights. Additionally, they all share the same trainable dimensionless features.

When the model is pretrained, the FP classifier is no longer updated in the training process with field data. When training

TABLE 10 Comparison of model configurations and predicted field pressures.

Model configuration			Predicted pressure MAE (10^6 Pa)			
K_α and ϕ_l^2	Dimensionless	Pretrain	Initialized	Pretrained	Trained	Tuned
Single MLP	No	Yes	4.38	3.18	1.78	1.72
Single MLP	No	No	4.38	-	2.35	2.26
Single MLP	Only	Yes	4.38	2.14	**1.12**	**1.04**
Single MLP	Only	No	4.38	-	1.56	1.40
Single MLP	Concatenated	Yes	4.38	2.14	1.12	1.06
Single MLP	Concatenated	No	4.38	-	1.54	1.38
MoE	No	Yes	4.38	2.46	1.65	1.60
MoE	No	No	4.38	-	2.49	2.47
MoE	Only	Yes	4.38	2.79	1.35	1.27
MoE	Only	No	4.38	-	2.06	2.09
MoE	Concatenated	Yes	4.38	2.32	1.32	1.25
MoE	Concatenated	No	4.38	-	1.90	1.85
MoE - no physics	Concatenated	Yes	13.24	42.35	1.15	1.14
MoE - no physics	Concatenated	No	13.24	-	1.85	1.77
Beggs and Brill	-	-	3.02	-	-	1.64

Bold indicates best metrics.

directly on field data without pretraining, the FP classifier weights are updated. However, no emergent flow pattern prediction capabilities were observed in this case. The initialization F_1 scores do not carry any meaning as they depend on the specific realization of the random weights initialization.

We now analyze how these models behave by evaluating the field metrics: the prediction error (MAE) of the pressures P_{wh}, P_{wf}, and temperatures T_{wh} and T_{wf}. The pressure MAE is shown in Table 10 for all model configurations.

The initialized models naturally performed worse than the Beggs and Brill correlation baseline. However, it is noticeable that the use of multipliers K_α and ϕ_l^2 improved the predictions compared to the model without multipliers (no physics), hinting at a possible benefit from the incorporated physical structure. Pretraining on experimental data consistently improved the performance of all models after they were trained with field data. This indicates that the initial training provided a strong physical starting point, enhancing the model's ability to generalize and adapt to the field dataset. The models that were pretrained achieved lower prediction errors than their counterparts trained only on field data.

The best performance was achieved by the physics-based Single MLP framework with only dimensionless inputs ($\text{MAE} = 1.12 \times 10^6 Pa$ after training and $\text{MAE} = 1.04 \times 10^6 Pa$ after tuning). The MoE framework, which yielded the best dP/dL metrics in the experimental dataset (Table 7), underperformed in the field dataset, showing that it overspecialized in the experimental data, and the learned flow patterns did not improve the field metrics. The model without physics also overspecialized during pretraining, which is reflected by the worse field metrics. The Beggs and Brill model, after adjusting the factors for frictional dP/dL, dT/dL and $HL = (1 - \alpha)$ within the interval [0.85, 1.15], substantially improved the pressure predictions but still did not match the neural network based models.

Finally, Table 11 shows the evolution of temperature prediction errors through the training pipeline. The pressure and temperature profiles are strongly coupled by the Joule–Thomson coefficient η_{JT}, leading to eventually worse temperature predictions when privileging pressure adjustments in the multi-objective loss function. This effect creates a trade-off: the tuning step slightly improved pressure predictions but worsened temperature predictions. In a system where most pipelines are insulated, the Joule–Thomson effect has more influence on temperature profiles than direct thermal exchange. Given this, the optimization process prioritized fitting the pressure data at the expense of temperature accuracy.

Ultimately, the balance between these metrics can be controlled by adjusting the weighting factor γ (Equation 21) to suit a specific engineering goal. However, the high sensitivity of the temperature predictions underscores the critical importance of the fluid's property model.

In contrast, the tuned Beggs and Brill benchmark performed well for both pressure and temperature. This is because its setup allowed for the application of a direct multiplicative correction factor to the calculated dT/dL, which can mask underlying flaws in the fluid property model.

To evaluate the trade-offs between each proposed architecture when evaluating both experimental and field data, we plotted

TABLE 11 Comparison of model configurations and predicted field temperatures.

Model configuration			Predicted temperature MAE (°C)			
K_α and ϕ_l^2	Dimensionless	Pretrain	Initialized	Pretrained	Trained	Tuned
Single MLP	No	Yes	3.62	3.35	3.46	4.45
Single MLP	No	No	3.62	-	3.57	4.98
Single MLP	Only	Yes	3.62	3.45	3.53	5.38
Single MLP	Only	No	3.62	-	3.49	5.88
Single MLP	Concatenated	Yes	3.62	3.44	3.53	4.98
Single MLP	Concatenated	No	3.62	-	3.49	6.20
MoE	No	Yes	3.62	3.33	3.50	4.53
MoE	No	No	3.62	-	3.47	2.72
MoE	Only	Yes	3.62	3.49	3.50	6.00
MoE	Only	No	3.62	-	3.42	4.56
MoE	Concatenated	Yes	3.62	3.46	3.44	6.30
MoE	Concatenated	No	3.62	-	3.41	4.33
MoE - no physics	Concatenated	Yes	4.67	20.84	3.47	**2.36**
MoE - no physics	Concatenated	No	4.67	-	3.38	5.49
Beggs and Brill	-	-	3.87	-	-	2.71

Bold indicates best metrics after tuning.

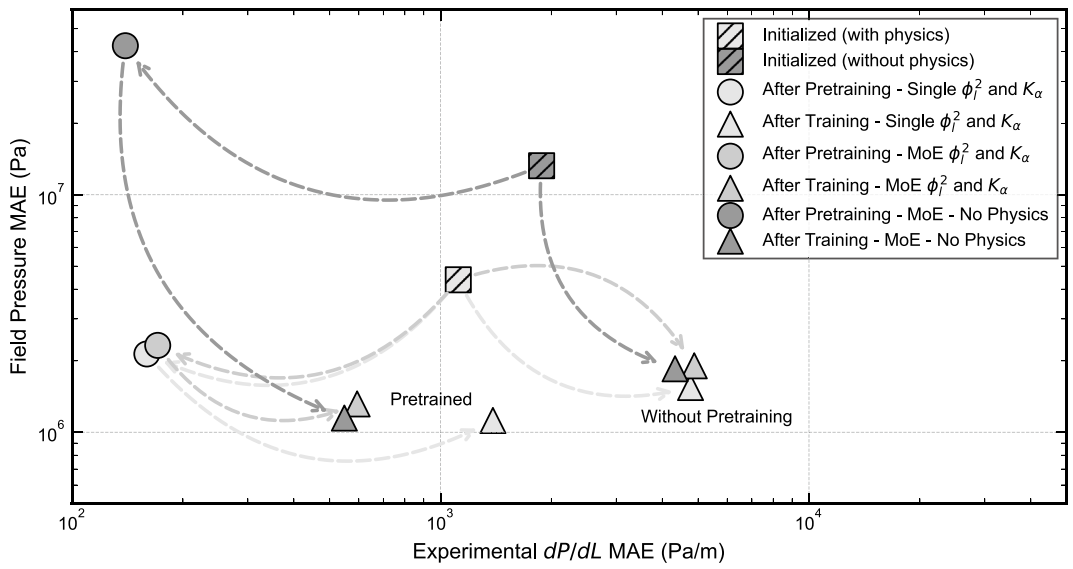

FIGURE 7
Trade-off between experimental and field test metrics.

Figure 7 showing both the experimental dP/dL and field P error predictions (MAE) based on the models relying on concatenated dimensionless features, which were the best overall performers across all the metrics. Better metrics are located in the lower-left portion.

The models based on the K_α and ϕ_l^2 multipliers, when initialized, yielded better metrics in both experimental and field data than the model initialized without physics. The latter, however, displayed outstanding performance in the experimental metrics after pretraining, as well as in the filed metrics after training directly

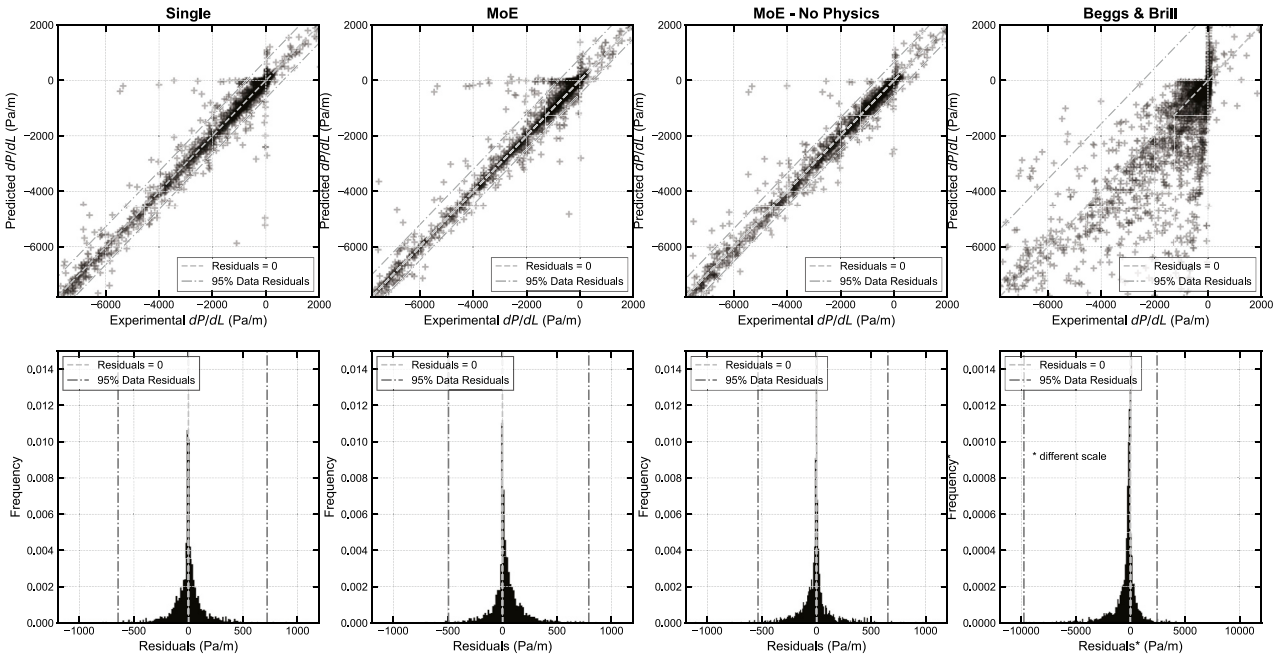

FIGURE 8
dP/*dL* predictions and residuals across different modeling approaches.

on field data (without pretraining). However, one good metric is worse than the other (see the large spread of the possible training paths for the model without physics). This leads us to conclude that this architecture overspecializes in one or other training objectives because it is not generalist.

As shown in Figure 7, the training paths including pretraining consistently achieved superior metrics on the field dataset. This demonstrates that initializing the model with a foundational understanding of flow physics from the experimental data allowed for more effective fine-tuning and resulted in a more accurate final model. We can also see that based on the specific engineering objective, we are subject to a trade-off between the two domains. The choice of architecture and underlying complexity is dependent on the model's application. Furthermore, the MoE framework offers significant qualitative advantages that make it more suitable for engineering applications. The tangible benefit of this configuration is the interpretability of having a flow pattern prediction, thus enabling engineering analysis for design purposes (e.g. avoiding instabilities for better process control) and also troubleshooting capabilities, as the dP/dL and α calculations are dependent on the flow pattern.

3.2 Pretraining with the experimental dataset

Following the quantitative results from the previous section, we will now qualitatively assess the model's prediction capabilities when pretrained on experimental data. Figure 8 displays the predicted values (upper row) for dP/dL and the respective error residuals (lower row), considering the following model configurations: single MLPs for K_α and ϕ_l^2; MoE framework for K_α and ϕ_l^2; MoE framework without multipliers (no physics); Beggs and Brill baseline. The visualized metrics from the trainable models (first three) were trained using concatenated dimensionless inputs—the best input features according to the comparison on Section 3.1.

In the single MLP model, the scatter plots illustrate a dense concentration of data points around the identity line, which is indicative of a strong model fit. The residuals distributions (bottom row) show that the modes are close to zero and the tails are mostly symmetrical, which means that the model's predictions are unbiased. For the MoE model, although the scatter plots indicate good fitness and the metrics are superior to the single MLPs' metrics, the dP/dL predictions are biased, as the asymmetrical tail in the residuals distribution indicate. The model shows a tendency of overestimating dP/dL. The distribution for α predictions, however, is unbiased for both single and MoE physics-based models.

The MoE framework without the multipliers (no physics) showed similar fitness and metrics in the dP/dL predictions. The residuals plot, however, shows a thicker tail in the left, meaning that the model is biased and has a tendency of underestimating dP/dL. The Beggs and Brill predictions are shown for illustration only, as their residuals are larger than the other models', and the plotting axes had to be adjusted.

The α predictions (Figure 9) show good fitness (predictions close to the identity line) for all the physics-based (MoE and single MLP) models. The MoE framework without multipliers (no physics), however, shows worse fitness, as shown in the scatter plot, and a bias toward the right in the histogram, showing a tendency of overestimating α values. Beggs and Brill displays large residuals also for α predictions.

For both dP/dL and α predictions from the trainable models, the presence of some outliers can be noted. Taking into account that the

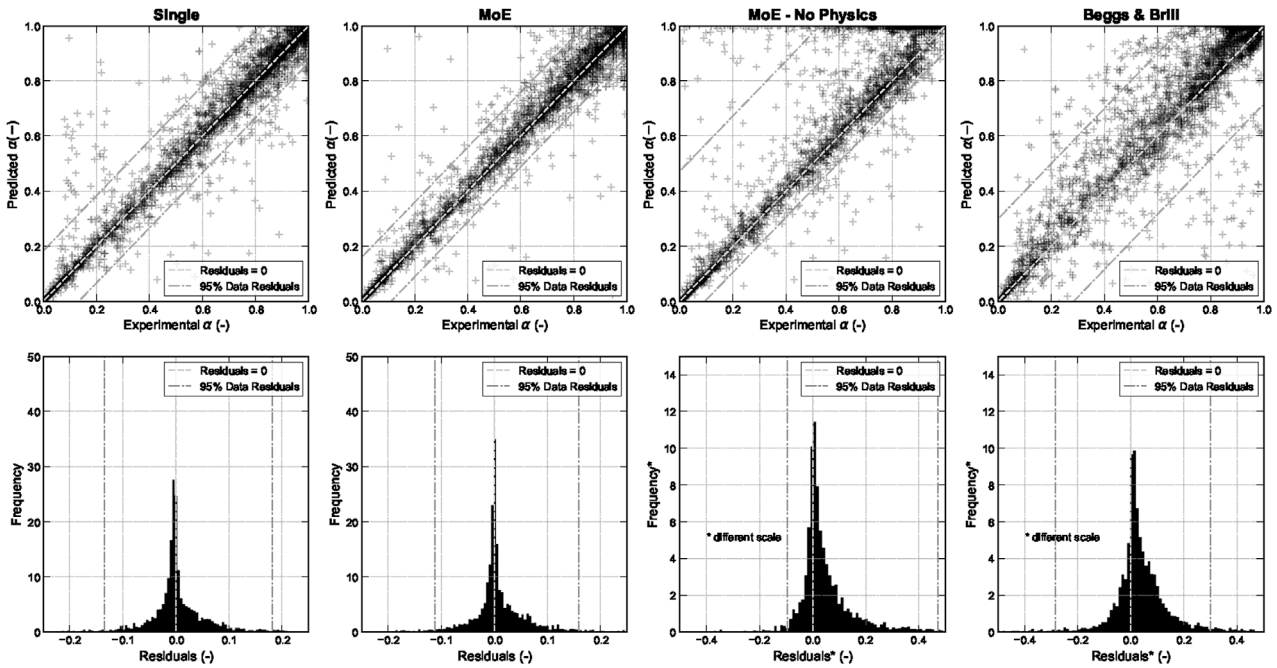

FIGURE 9
α predictions and residuals across different modeling approaches.

TABLE 12 Flow pattern predictions confusion matrix (test dataset).

True label	Model				Barnea (1987)			
	S	A	I	B	S	A	I	B
Stratified	**773**	17	32	9	**608**	180	41	2
Annular	20	**518**	28	1	60	**399**	108	0
Intermittent	34	31	**1413**	37	88	280	**1096**	51
Bubble	2	2	21	**273**	20	11	125	**142**

Bold indicates predicted flow pattern = true flow pattern.

test dataset has 3,211 data points, the number of outliers is not significant. No correlation was observed between the larger residuals and any specific flow pattern or pipe inclination range.

Considering the model with the MoE architecture, the flow pattern predictions for the test dataset can be compared with the model of Barnea (1987) in the confusion matrix in Table 12. Our model resulted in a F_1 score of 0.927, while Barnea's resulted in a F_1 score of 0.699.

For a specific configuration of fluid characteristics in a horizontal pipeline of the field data test cases, the model's predicted flow pattern map (considering the highest p_i for each point) compared to Barnea's model map can be visualized in Figure 10.

While the overall topology of the predicted flow pattern map shows a strong resemblance to the Barnea baseline, its deviations highlight limitations inherent in a purely data-driven classification approach, especially when dealing with data scarcity. The model successfully captures the main physical transitions from the large, stratified region to intermittent flow with increasing liquid velocity and to annular flow with increasing gas velocity. However, a significant limitation appears at high gas and liquid velocities, a region corresponding to churn flow (part of our intermittent class), where the model erroneously predicts bubble flow. This misclassification is directly attributed to the underrepresentation of data points for the churn regime in our experimental dataset.

3.3 Effect of learnable dimensionless numbers

While exploring good combinations of hyperparameters for the models, the effect of the number of dimensionless numbers was also investigated. Across multiple neural network configurations (widths and depths), the use of learnable dimensionless numbers consistently improved the metrics, lowering the average mean absolute errors (MAE) for the validation dP/dL and α and increasing the flow pattern classification F_1 score, as shown in Figure 11, where these metrics were compared to those without the use of dimensionless numbers (MAE_{nd} and $F_{1,nd}$).

This finding is of considerable importance as it provides strong empirical validation for the use of learnable dimensionless numbers as a superior feature engineering strategy. Instead of relying on predefined, universal dimensionless groups, the BuckiNet framework allows the model to autonomously discover the most relevant physical relationships directly from the data. This data-driven approach enables the identification of parameter groupings that are optimally tuned for the specific fluid properties and flow conditions of the dataset, enhancing the model's physical consistency and predictive power. The clear trend in Figure 11, where performance improves and then saturates, indicates that the model finds an optimal basis of dimensionless features, effectively

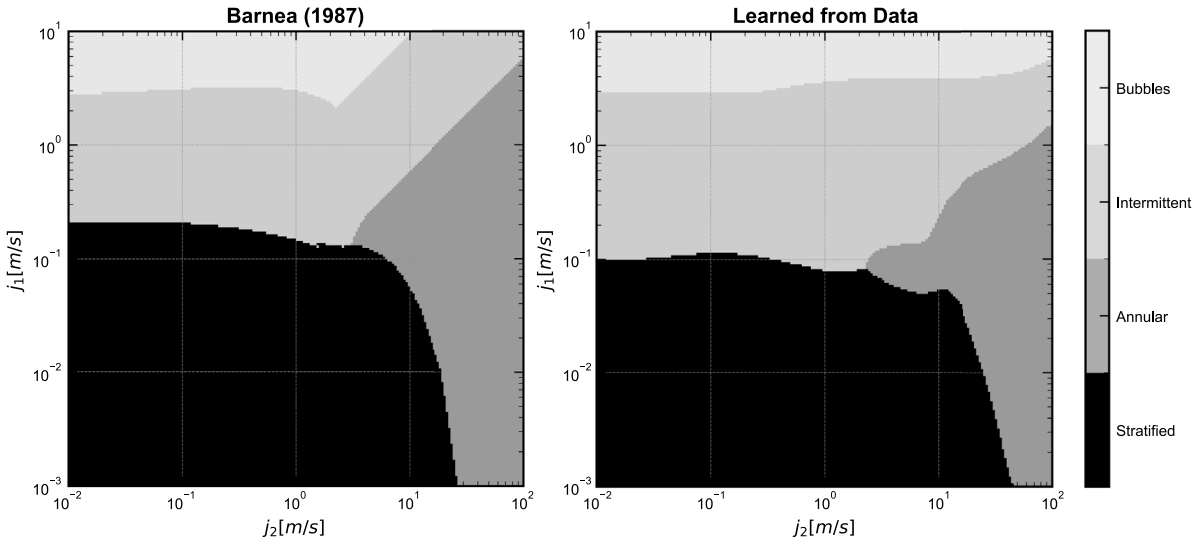

FIGURE 10
Example of a predicted flow pattern map compared to the result obtained using Barnea's method, grouping the flow patterns in the four classes described in Section 2.5.

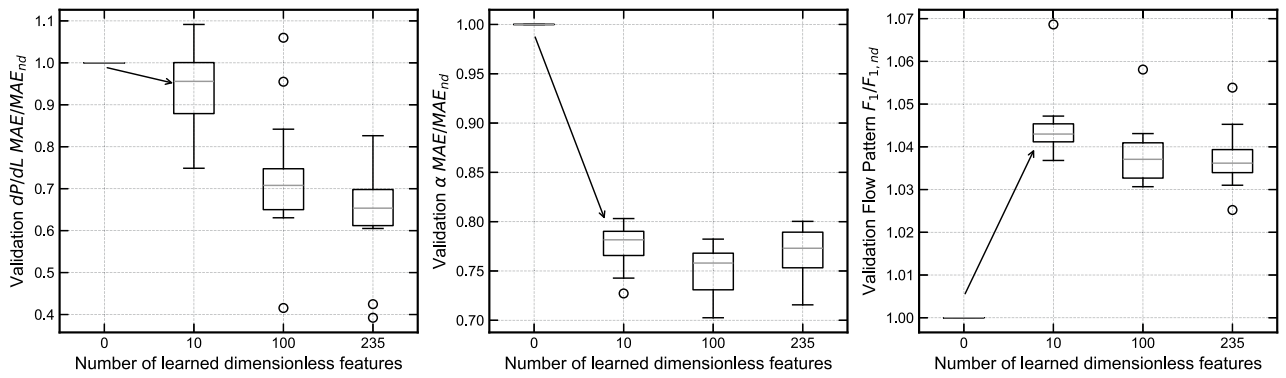

FIGURE 11
Effect of the number of learned dimensionless features on the predicted dP/dL and α mean absolute errors and flow pattern prediction F_1 score in the validation set.

capturing the essential physics while avoiding unnecessary complexity. This automated discovery of a physically meaningful feature space represents a key contribution of our framework, offering a more robust and generalizable alternative to manual feature engineering.

To illustrate the learned soft dimensionless features, we took the ten features with the smallest residuals (null space loss) after pretraining the MoE model with experimental data, and related their exponents (which are the elements of the matrix Φ_p) in Table 13. The residuals in the last column are the sum of the absolute values of the residuals in the mass (M), length (L), and time (T) dimensions. Note that θ is already dimensionless (rad), but it was left as part of the non-dimensionalization process to contribute to the feature engineering.

The soft dimensionless features were analyzed to identify known dimensionless from the literature, aggregating the velocities ($j_1^a \cdot j_2^b = j^{a+b}$), the densities ($\rho_1^a \cdot \rho_2^b = \rho^{a+b}$), and the viscosities ($\mu_1^a \cdot \mu_2^b = \mu^{a+b}$). For example, the soft dimensionless number π_2 can be written as in Equations 31, 32:

$$\pi_2 = j_1^{-0.007} \cdot j_2^{0.006} \cdot \rho_1^0 \cdot \rho_2^{-0.145} \cdot \mu_1^{0.223} \cdot \mu_2^{0.070} \cdot \sigma_{21}^{-0.147} \cdot d^{0.098} \cdot \epsilon^{-0.227} \cdot \theta^{0.192} \tag{31}$$

$$= (\mathrm{Re})^{-0.292} \cdot (\mathrm{We})^{0.146} \cdot (\varepsilon/d)^{-0.230} \cdot j^{-0.002} \cdot \rho^{0.001} \cdot \mu^0 \cdot \sigma_{21}^0 \cdot d^{0.014} \cdot \epsilon^{0.003} \cdot \theta^{0.192}, \tag{32}$$

where $\mathrm{Re} = j \cdot \rho \cdot d \cdot \mu^{-1}$ is the Reynolds number, $\mathrm{We} = j^2 \cdot \rho \cdot d \cdot \sigma_{21}^{-1}$ is the Weber number, and ε/d is the pipe's relative roughness. Identifying these dimensionless numbers leaves residuals with significantly smaller exponents in the dimensional quantities.

TABLE 13 Sample exponents and residuals for learned soft dimensionless features.

	Exponent										Residual
π_i	j_1	j_2	ρ_1	ρ_2	μ_1	μ_2	σ_{21}	d	ϵ	θ	$\sum \lvert r \rvert$
π_1	0.035	0	0	−0.020	0.041	0.049	−0.063	0	0	−0.087	0.014
π_2	−0.007	0.006	0	−0.145	0.223	0.070	−0.147	0.098	−0.227	0.192	0.015
π_3	−0.015	−0.049	0.047	−0.035	0	−0.083	0.073	0.231	−0.199	−0.037	0.019
π_4	0	0	0	0.093	−0.135	−0.031	0.082	0	0.104	−0.026	0.020
π_5	−0.074	0	0	−0.119	0.034	0.135	−0.055	0	−0.109	−0.090	0.025
π_6	0	−0.100	0.033	−0.068	−0.021	0	0.053	−0.017	0	0.207	0.026
π_7	0	−0.010	0	0	0.019	0	0	−0.042	0.075	0.078	0.031
π_8	0.059	−0.168	−0.129	−0.007	0	0.180	−0.027	−0.010	−0.108	−0.133	0.035
π_9	−0.134	0.090	0	−0.053	0.094	0	−0.018	−0.020	0	−0.153	0.038
π_{10}	0.091	0	0	0.040	−0.051	0.041	−0.031	0	0	0.108	0.038

Although further exploration can be performed to identify and catalog known dimensionless numbers from the literature which eventually emerge from our model, it was shown that, besides the benefits in the model quality as measured by the metrics, meaningful physical information can be extracted from these features.

3.4 Training with field data

As established in the training strategy comparison (Section 3.1), pretraining the model with experimental data enhances its performance on field data. Although a significant domain gap exists between experimental and field conditions, as will be discussed in Section 4, our transfer learning approach successfully bridges this gap.

Considering the best performing model during pretraining, using the MoE approach with concatenated dimensionless numbers and training with field data without parameter tuning (as the tuning affected the temperature prediction metrics), we obtained a MAE of 1.32×10^6Pa in the predicted pressure and 3.44 °C in the predicted temperature (Tables 10, 11). All the predicted pressures and temperatures in the test dataset are visualized in Figure 12.

The dash–dotted lines comprehend an interquartile range of 95% of the prediction errors (residuals). Good fitness is observed in the downhole pressures P_{wf}, as most predictions are close to the identity line (dark dashed line). The Xmas tree pressures P_{wh} are harder to predict via simulation, given topside pressures P_{ck}, due to the longer distances and mixed inclinations (flowlines and risers) compared to the flow in the wellbore (P_{wf} predictions), accumulating larger uncertainties. The predicted temperatures T_{wf} and T_{wh} show larger relative errors than the pressures, resulting in a more spaced interquartile range.

Evaluating the baseline Beggs and Brill model, after tuning the correction factors we obtained a MAE of 1.64×10^6Pa in the predicted pressure and 2.71 °C in the predicted temperature (Tables 10, 11). The predictions are shown in Figure 13.

The overall pressure prediction errors from the Beggs and Brill correlation are similar to our model's errors. The temperatures were better predicted, mainly due to the correction factors applied directly on the calculated dT/dL—a common practice in engineering, employed in this example to simulate a real field simulation fitting process.

Figure 14 visualizes the complete pressure and temperature profiles for a particular production test from the test dataset. The discontinuities shown at the Xmas tree occur because the well and pipeline segments are simulated separately. Providing the complete geometry to the model at once would produce contiguous profiles but also lead to the accumulation of prediction errors.

This particular result does not indicate that the MoE framework yields better simulations than Beggs and Brill; the metrics do. Figure 14 demonstrates the capability of calculating entire profiles from a model trained only on sparse field data, without using any previously programmed correlation. The simulated profiles have a wide range of applications in industry, from design to production monitoring and flow assurance.

3.5 Parameter tuning

The parameter tuning step, performed after model training, was set to adjust three key sets of parameters: the roughness of all pipes, their global heat transfer coefficients, and the linear transformations applied to the Joule–Thomson coefficient.

Figure 15 (left) represents the relative change in the absolute roughness ε for each class of pipe (well tubings, steel catenary risers, and flexible flowlines and risers) for every trained model. The tuning process systematically alters the pipe roughness values and minimizes the objective function (Equation 21). For most well tubings, the optimization process reduced the initial roughness values by over 2% to minimize the objective function. This adjustment indicates that the baseline model overestimates pressure loss in the well tubings. Most flexible flowlines and risers, on the other hand, did not require major adjustments, while the steel catenary risers had their roughness values increased by almost 2%.

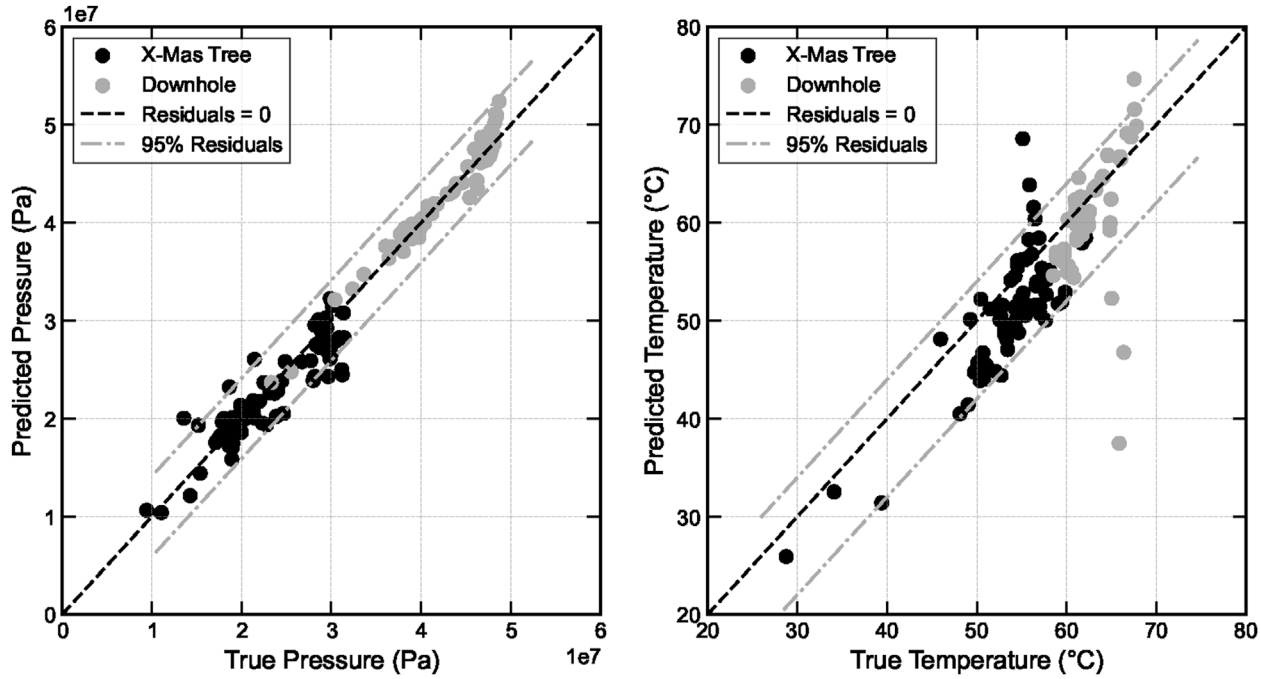

FIGURE 12
Pressure and temperature predictions from the pretrained and trained model (MoE framework, without tuning) in the test dataset.

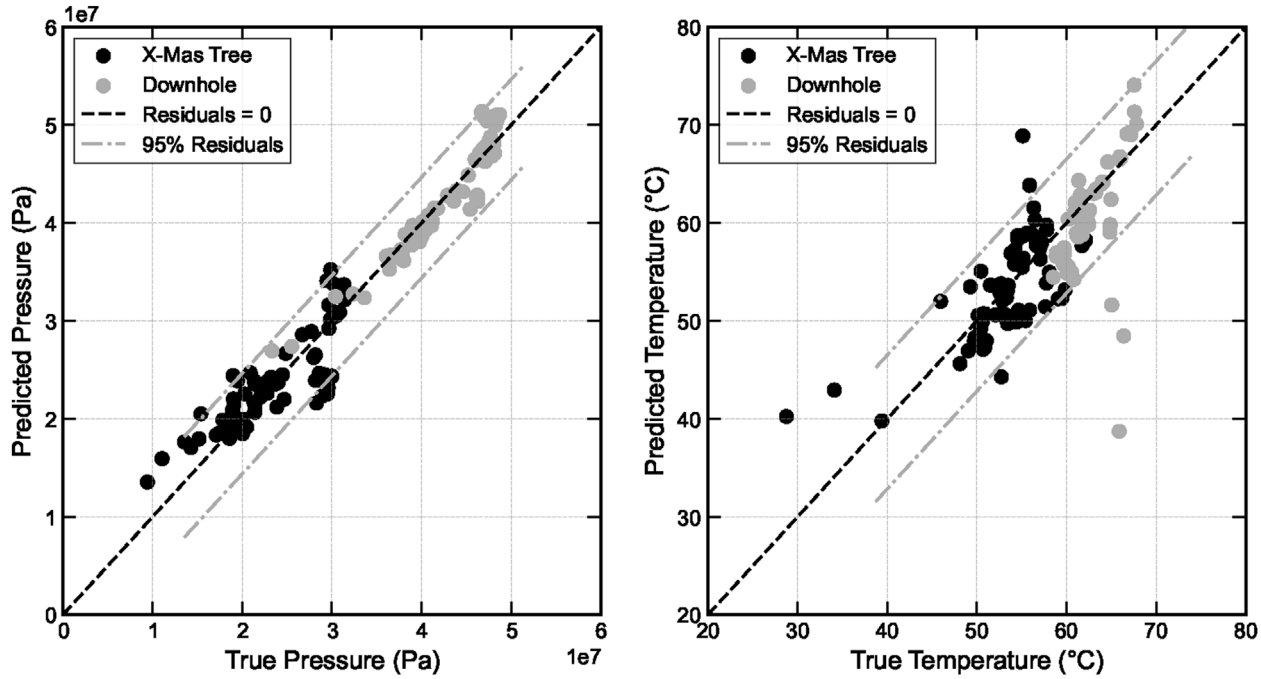

FIGURE 13
Pressure and temperature predictions from the Beggs and Brill correlation (tuned with field data) in the test dataset.

The global heat exchange coefficient U, which was also tuned in the process, and the tuning curves are also shown in Figure 15 (right). The major adjustments were made in the flexible flowlines (reduced U) and steel risers (increased U), while minor adjustments were performed in the well tubings and flexible risers.

Furthermore, Figure 16 shows the tuning of the Joule–Thomson linear transformation coefficients a_{JT} and b_{JT}

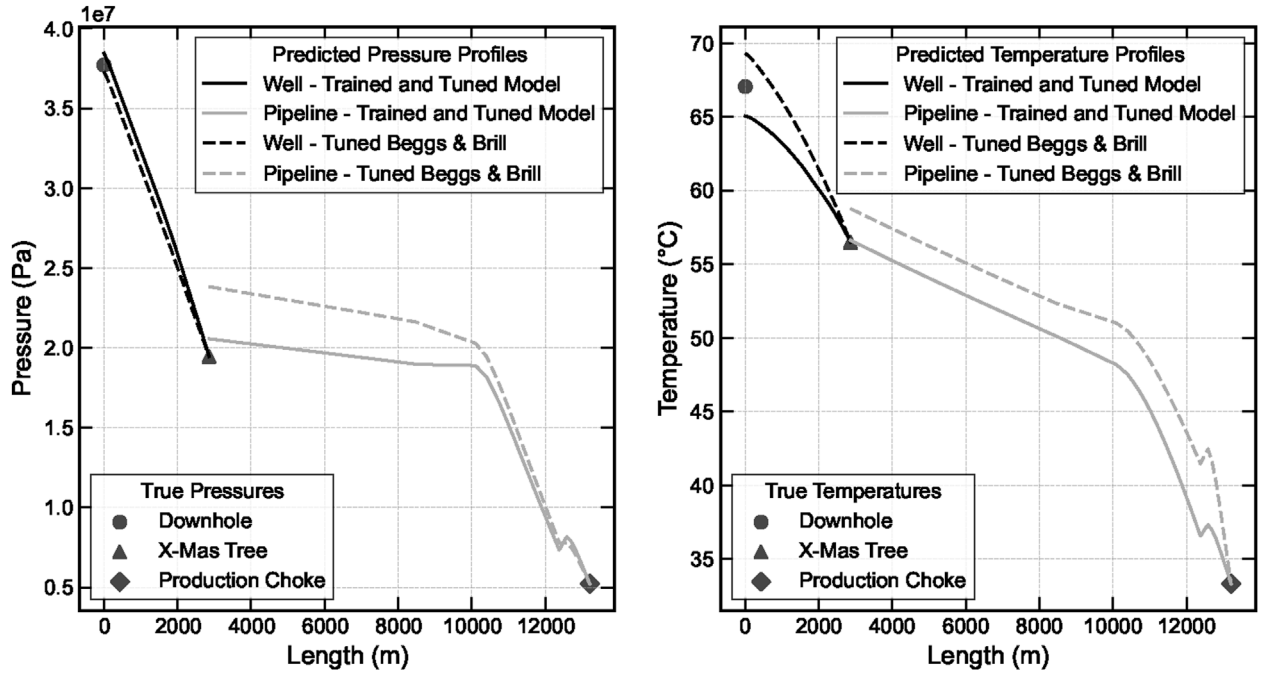

FIGURE 14
Predicted pressure and temperature profiles for an instance of the test dataset. Comparison between the tuned Beggs and Brill correlation and the MoE framework model trained and tuned with field data, without pretraining.

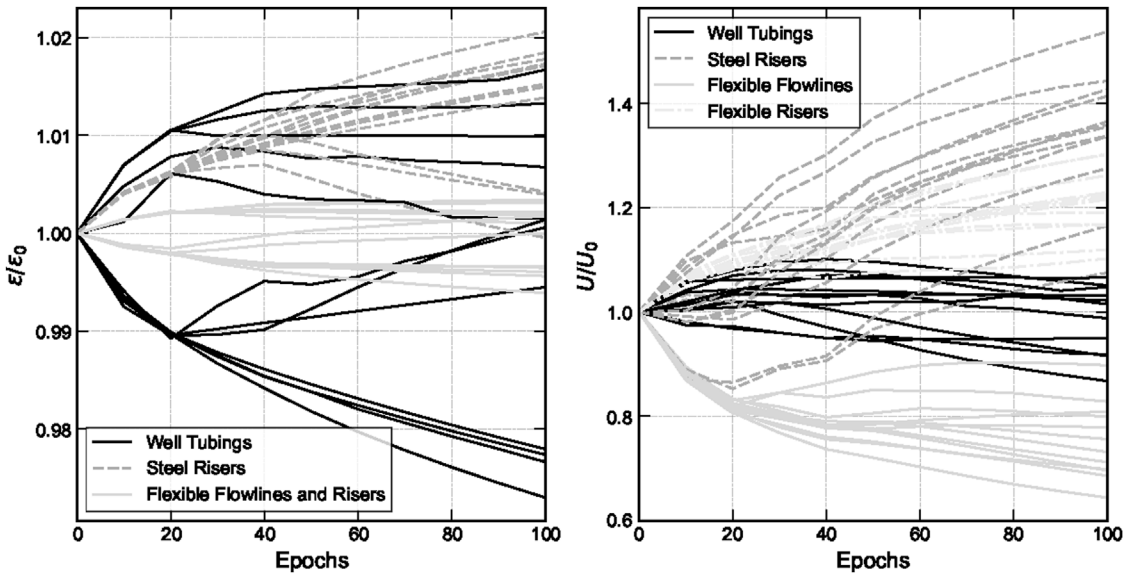

FIGURE 15
Roughness (ε) and heat exchange coefficient (U) tuning process in all wells and pipelines relative to their initial values ε_0 and U_0.

from Equation 8. Most adjustments were made in the bias a_{JT}, initialized as 0, to account for the changes needed in η_{JT} to further minimize the training and tuning objective (Equation 21). The multiplicative coefficient b_{JT}, which was initialized as 1, was less updated in the tuning process due to its smaller influence on η_{JT}, which is roughly within the interval $[-5 \times 10^{-7}, 2 \times 10^{-6}]$.

As seen in both ε and η_{JT} tuning curves (Figures 15, 16), convergence was not achieved, which means that the optimization could continue to improve the metrics even further.

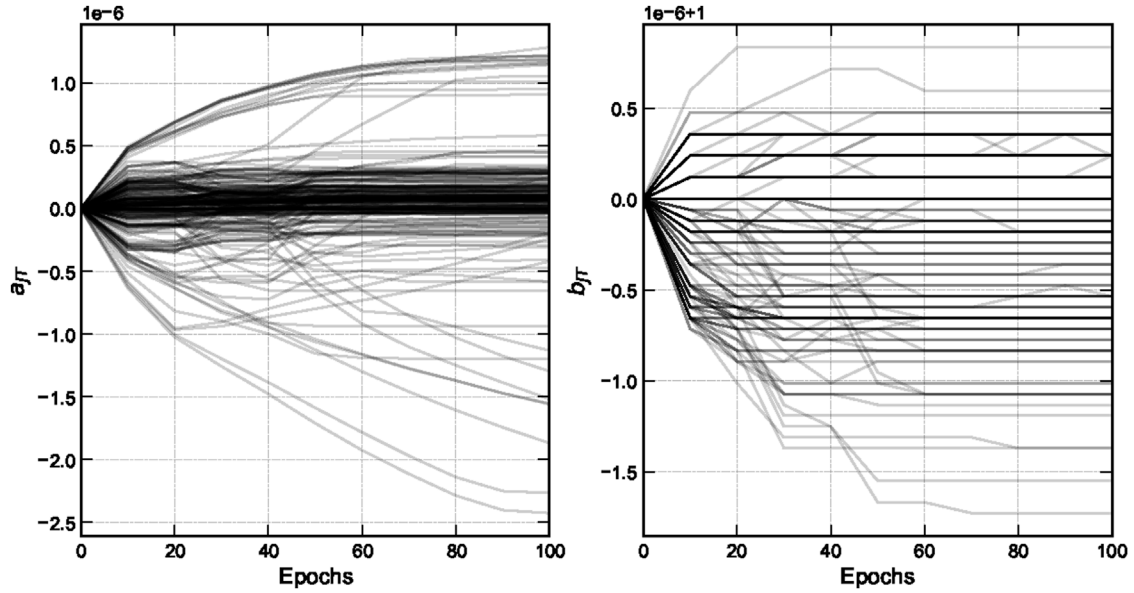

FIGURE 16
Transformation coefficients a_{JT} and b_{JT} for the Joule–Thomson coefficient η_{JT} for each well in the tuning process.

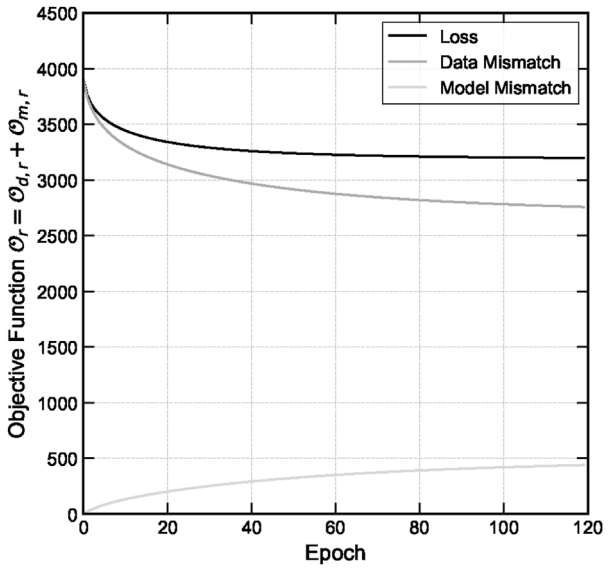

FIGURE 17
Convergence from the RML objective function and its components (model mismatch and data mismatch).

However, in our experiment, the number of epochs was limited to 100.

Tuning physical parameters can improve specific metrics, such as the pressure MAE (Table 10), but this requires careful balancing. As shown in Table 11, these adjustments can simultaneously degrade other metrics like temperature, revealing a trade-off between competing objectives. Furthermore, since the influence of different parameters can vary by orders of magnitude, the employment of individual learning rates for each type of parameter is required to manage these sensitivities more effectively.

3.6 Bayesian inference - RML

The following scenario was created to test the RML algorithm capabilities and its coupling with the differentiable model: given a production test, we want to simulate the sub-sea and downhole pressures and temperatures P_{wh}, P_{wf}, T_{wh}, and T_{wf} and compare those with actual measurements to perform flow assurance diagnostics. We will allocate the model's inputs (the topside pressure P_{ck} and temperature T_{ck}) and the model's parameters (liquid flow rate Q_{liq}, water cut WC, and the gas–oil ratio GOR) under the vector $\vec{m}$ (Equation 33). This specific well does not operate with gas-lift.

$$\vec{m} = \left[P_{ck}, T_{ck}, Q_{liq}, WC, GOR\right]^T. \tag{33}$$

The data observed $\vec{d}_{obs}$ (Equation 34) are the downhole pressure P_{wf} and temperature T_{wf} and the Xmas tree pressure P_{wh} and temperature T_{wh}, which are also calculated by the model $(g(\vec{m}))$.

$$\vec{d}_{obs} = \left[P_{wf}, T_{wf}, P_{wh}, T_{wh}\right]^T. \tag{34}$$

The prior distributions for P_{wf}, T_{wf}, P_{wh}, and T_{wh} were modeled considering the variance of the model prediction error, and it was considered a 3% error within three standard deviations for the other variables. Cross-correlations were not taken into account, so $\mathbf{C}_m$ and $\mathbf{C}_e$ are diagonal matrices.

The optimization process converges smoothly (Figure 17). The reduction on the data mismatch $\mathcal{O}_{r,d}$ represents the model's capability to explain the observed data, while the increasing model mismatch $\mathcal{O}_{r,m}$ represents the deviation from the model's

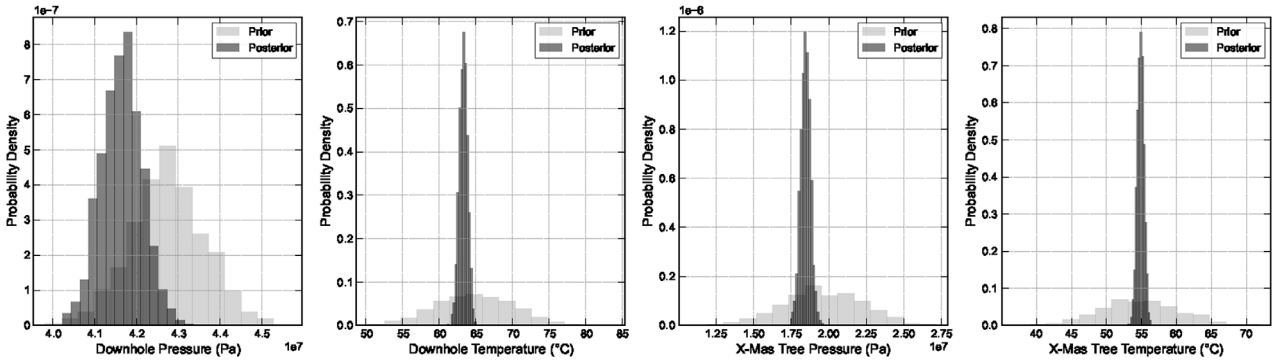

FIGURE 18
Prior and posterior distributions from the predicted pressures and temperatures.

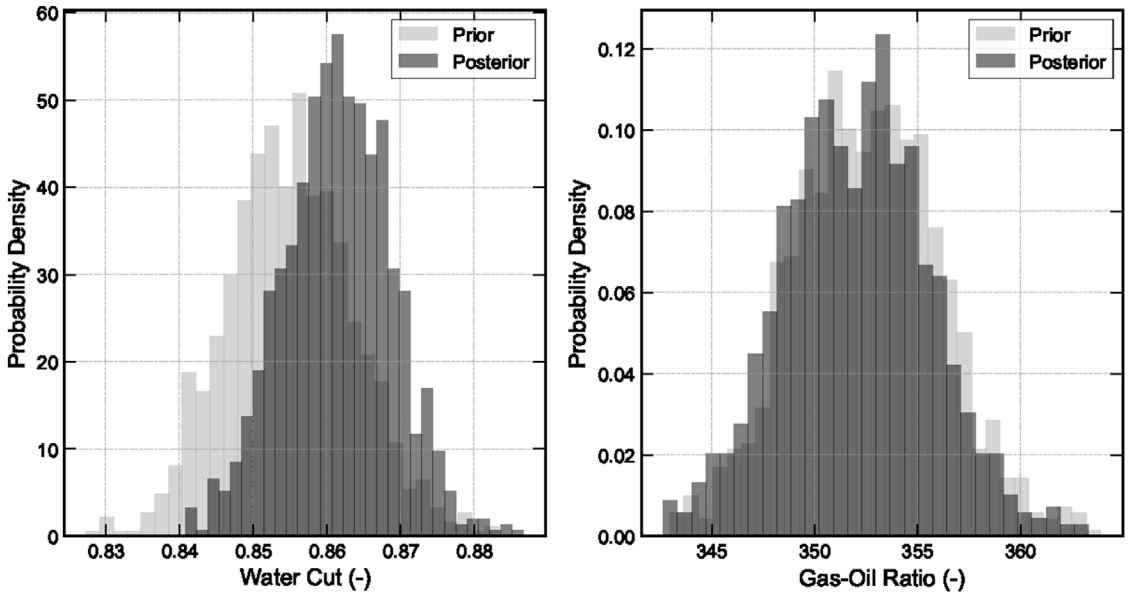

FIGURE 19
Prior and posterior distributions from the model's *WC* and *GOR*.

parameters from their prior distribution, acting as a regularization term in the overall objective function $\mathcal{O}_r$. This process results in narrower posterior distributions for the simulated data compared to their prior distributions (Figure 18). The posterior downhole pressure P_{wh} presents the largest uncertainties compared to the other variables as a result of the disagreement between observed and simulated values. This is in addition to the accumulated simulation error obtained by simulating the entire profile at once (calculating P_{wf} and T_{wf} given P_{ck} and T_{ck}).

The RML process updates the model parameters $\vec{m}$ to maximize their posterior probability, which is a combination of their prior probability distribution and the likelihood of the observed data $\vec{d}_{obs}$ (Equation 29). Comparing the parameters' prior and posterior distributions, we can get insights into the parameter changes that may influence the deviations on the observed data. Figure 19 shows the distribution for the water cut *WC* and gas–oil ratio *GOR* parameters. While there was no noticeable shift in the *GOR* distribution, the *WC* visibly shifted to the right, increasing the mean from 0.855 to 0.862.

This distribution shift in the *WC* parameter means that an increasing *WC* could explain the observed deviations from Figure 18. Furthermore, comparing the sample's *WC* prior and posterior distributions, we can calculate the probability of *WC* having increased: $P(WC_{posterior} > WC_{prior}) = 72\%$.

The resulting pressure and temperature profiles with incorporated uncertainties can also be visualized (Figure 20), based on the simulations using the posterior distribution for m. The central dark lines indicate the mean of the posterior simulated pressure and temperature profiles, while the shaded areas represent the confidence intervals ($\pm 3\sigma$). This visualization of uncertainties in the simulation results is a powerful feature enabled by our method.

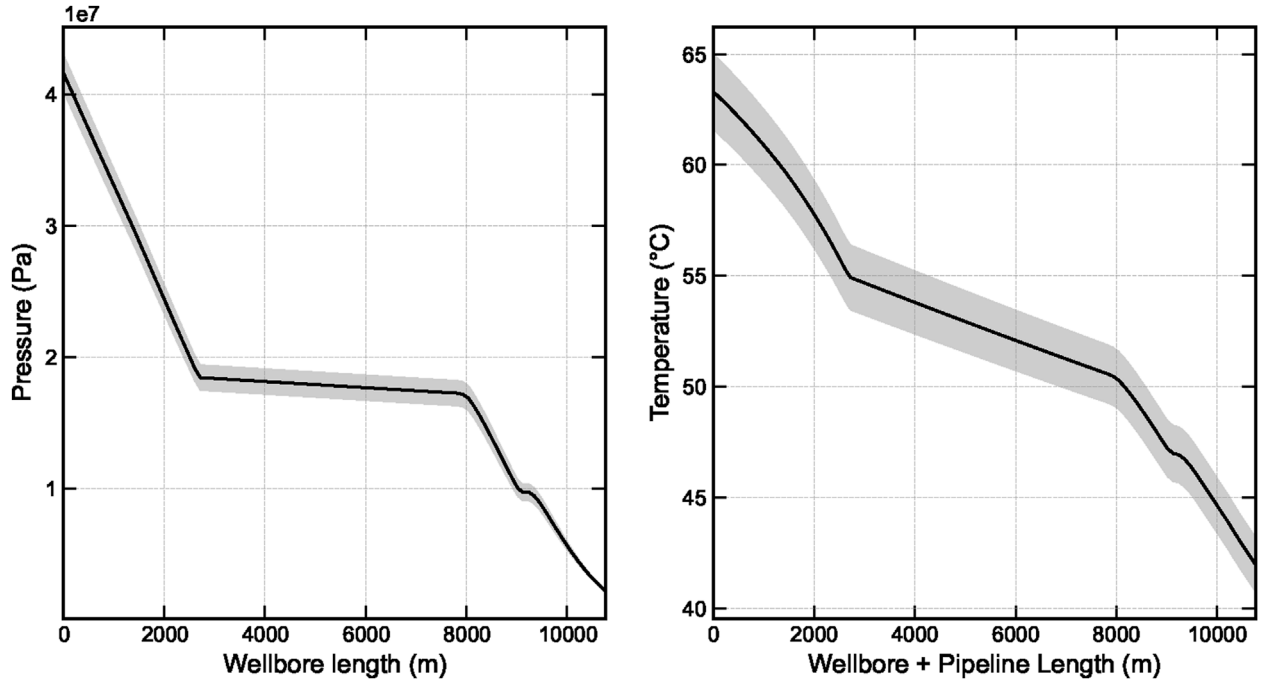

FIGURE 20
Pressure and temperature profile uncertainties ($\pm 3\sigma$) considering the posterior parameters distribution.

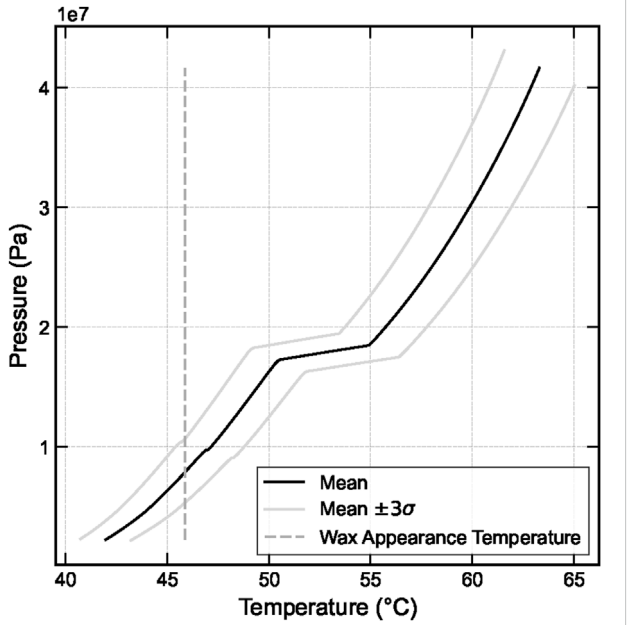

FIGURE 21
Thermohydraulic profile with uncertainties ($\pm 3\sigma$) compared to the fluid's wax appearance temperature.

By accounting for uncertainty, the simulated pressure and temperature profiles enable more reliable operational diagnosis and informed system design in flow assurance. More robust analysis can be done regarding the probability of hydrate formation or wax deposition in steady-state production. In the example shown in Figure 21, the mean P vs. T profile is plotted along with the posterior uncertainties ($\pm 3\sigma$). The superposition with the wax appearance temperature (WAT) indicates a high probability of crystallization, as the entire uncertainty range crosses the WAT line ($T \leq \text{WAT}$).

There is a wide range of applications for the use of Bayesian inference with the randomized maximum likelihood (RML) method using steady-state simulations for multiphase flow: nodal analysis with uncertainties, calculating the probability of blowout, virtual sensing, robust flowrate estimation, and anomaly detection, to name a few. This section demonstrated the method's capability and simple usage, enabled by the differentiable framework developed.

4 Discussion

As initially hypothesized, pretraining the model proved to be a highly effective strategy for improving its performance on field data. Our results confirm that establishing a baseline understanding of flow physics on a broad experimental dataset allows for more robust and accurate fine-tuning on sparser, application-specific field measurements. The use of trainable dimensionless features and a physics-informed structure were key enablers of this successful knowledge transfer.

In order to analyze the differences between the experimental dataset and the *in situ* conditions observed in the field, the entire field dataset was simulated, then all the input features were compared to the experimental dataset in a t-SNE plot (van der Maaten and Hinton, 2008) (Figure 22). In the (a) plot, only the

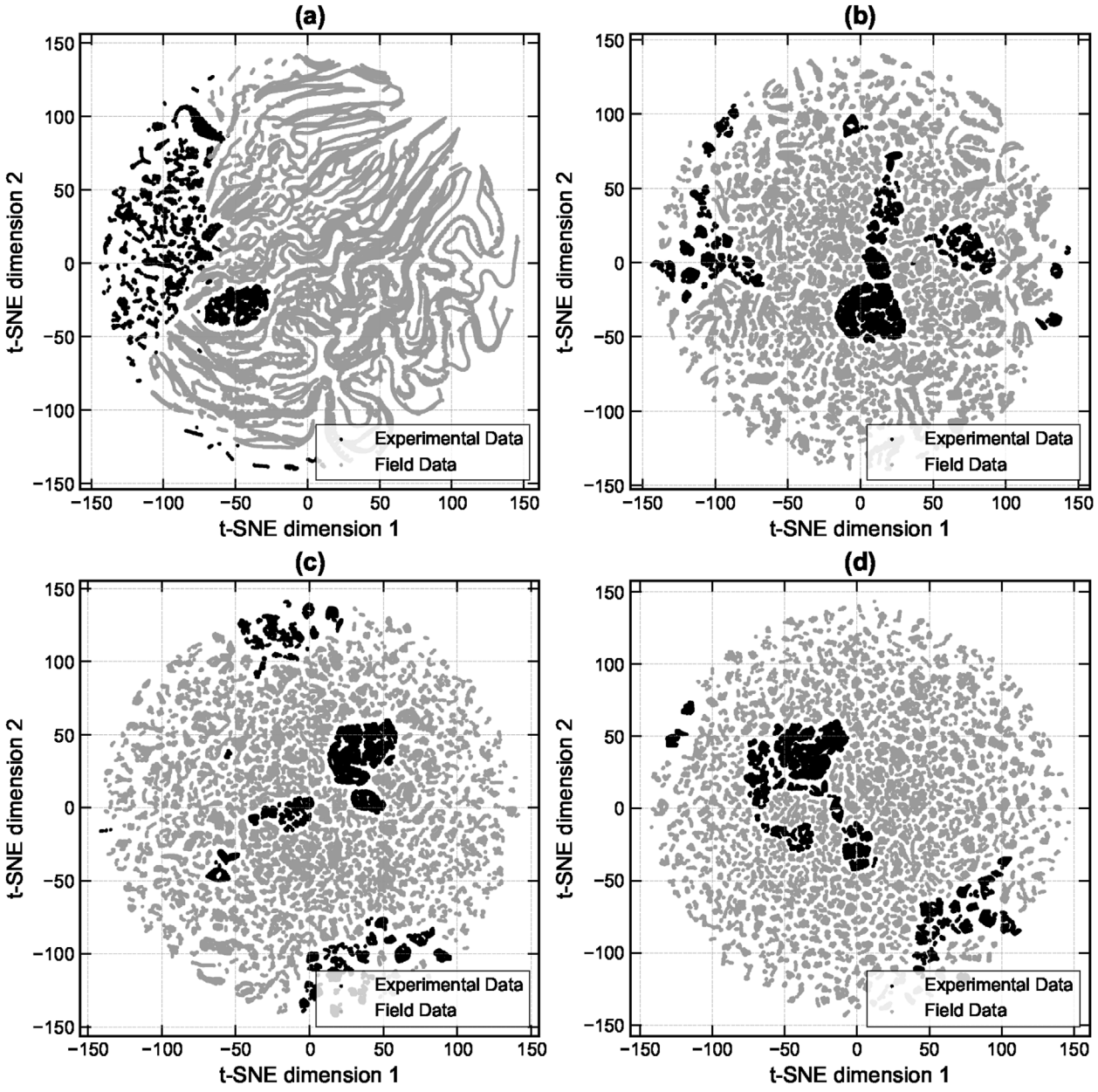

FIGURE 22
t-SNE plot from the model's input features when evaluating both experimental and field datasets, considering: **(a)** dimensional features only; **(b)** normalized concatenated dimensional and dimensionless features in a pretrained model (only with experimental data); **(c)** normalized concatenated dimensional and dimensionless features in a pretrained (with experimental data) and trained (with field data) model; **(d)** normalized concatenated dimensional and dimensionless features in a model trained only with field data (not pretrained).

dimensional features were considered, which is why the field data display this "snake" pattern. There is no overlap between the field and the experimental data. The (b), (c), and (d) plots represent the data after normalization, with dimensionless numbers. There is practically no overlap between the sets in those cases, except for a few small regions, possibly representing some generalization provided by the dimensionless numbers; however, this cannot be asserted purely based on this evidence.

The gap between the experimental and field data was known and, in fact, was a major motivation to developing models that are trainable with field data. The operating conditions in deep water are more extreme (higher pressures and temperatures) and the fluid compositions are quite different, containing contaminants such as CO_2 and high gas densities—conditions unseen on the experimental benches—which leads to a lack of data needed to properly model the closure relations in these scenarios. While the t-SNE analysis in Figure 22 clearly visualizes this domain gap, our results demonstrate that a transfer-learning approach can successfully bridge this gap. The model's ability to generalize suggests that pretraining, combined with its physics-informed architecture, allowed it to learn the underlying physical principles of two-phase flow rather than simply memorizing the conditions present in the experimental data.

On the other hand, there is room for improvement in the physical structure introduced in the modeling. Virtually any correlation or mechanistic model can be programmed using the automatic differentiation framework, allowing the training process to adjust any parameter of interest. It is expected that, with better physics-informed models, the residuals to be modeled by neural networks will be smaller. Ultimately, only a few coefficients from closure relations should be adjusted.

The implementation of more complex and implicit models, which require iterative root finding, poses challenges for three main reasons: the model needs to process batches, which will degrade the solver's performance; the need to propagate gradients based on the solution; mechanistic models show discontinuities (Shippen and Bailey, 2012), making it difficult for an optimizer to navigate the landscape and find convergence.

The process of developing a training procedure and the challenge of simultaneously matching pressure and temperature field measurements reveal another major issue: the high sensitivity of the fluid representation. Most engineers who perform production analysis and history matching rely on choosing a suitable flow correlation and then proceed to manually adjust tuning factors. The fluid model, although tuned to match the laboratory analysis in reservoir conditions, relies on equations of state (EoS) to extrapolate entire range of field operating conditions, and in most cases no adjustment is made on the fluid model during the well life cycle. This presents an opportunity to extend our differentiable framework to also update fluid parameters during training with field data.

5 Conclusion and future work

This study successfully developed and validated a novel end-to-end differentiable framework for modeling steady-state two-phase flows. By integrating a neural ODE formulation with physics-informed neural networks, the proposed model accurately predicts pressure and temperature profiles along complex deepwater well and pipeline geometries. The results demonstrate that our approach, when trained and tuned directly with field data, achieves better accuracy in pressure prediction than the proposed tuned Beggs and Brill (1973) correlation benchmark. The model's differentiability proves to be a powerful feature, enabling not only the optimization of neural network components but also the automated tuning of physical parameters like pipe roughness, effectively creating a self-calibrating simulation tool.

One of the most significant findings of this study is the successful application of a transfer learning approach for modeling real-world field conditions. We demonstrated that pretraining a model on a large experimental dataset and subsequently fine-tuning it with field data is a superior strategy to training solely on sparse field data. Our framework effectively bridges the domain gap between laboratory and field conditions, leveraging physics-informed structures and learnable dimensionless numbers to achieve high accuracy.

Furthermore, the utility of the differentiable framework extends beyond forward calculations. We demonstrated its seamless integration with advanced algorithms for inverse problems, using randomized maximum likelihood (RML) to efficiently perform uncertainty quantification. This capability highlights the model's potential as a cornerstone for digital twin applications, enabling robust history matching, virtual sensing, and what-if scenario analysis with quantifiable confidence intervals.

Despite these promising results, this study also identifies paths for future research. The results of the simulations are highly sensitive to the fluid's model quality. A key future direction is the development of a fully differentiable PVT module, which would allow the model to learn and adjust fluid thermodynamic properties or the coefficients from equations of state directly from field measurements. Additionally, the current framework is limited to steady-state conditions. Extending the model to handle transient multiphase flow is a natural and necessary next step to broaden its applicability to dynamic operational scenarios like startup and shutdown. Finally, further investigation into the physical interpretability of the learned dimensionless groups from the BuckiNet component could yield new insights into the dominant forces governing specific flow regimes encountered in the field.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material; further inquiries can be directed to the corresponding author.

Author contributions

AF: Writing – review and editing, Investigation, Writing – original draft, Software, Formal analysis, Data curation, Conceptualization, Methodology. SV: Investigation, Conceptualization, Supervision, Writing – review and editing, Formal analysis, Writing – original draft, Methodology, Validation, Data curation, Software. MC: Resources, Visualization, Funding acquisition, Project administration, Formal analysis, Methodology, Writing – review and editing, Supervision, Conceptualization.

Funding

The authors declare that financial support was received for the research and/or publication of this article. The authors acknowledge the technical support of Petrobras and the financial support of the company and ANP (Brazil's agency for national oil, natural gas, and biofuels) through the R&D levy regulation (Process 2017/00778-2). The authors declared that this work received funding from Petrobras. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, or the decision to submit it for publication.

Acknowledgements

Acknowledgments are also extended to Petrobras colleague Alexandre Anozé Emerick for insights on Bayesian inference

and to the Center for Energy and Petroleum Studies (CEPETRO), School of Mechanical Engineering (FEM), and Artificial Lift and Flow Assurance (ALFA) Research Group at UNICAMP. This project is the subject of a pending patent application (Castro et al., 2025b) and software registration (Castro et al., 2025a).

Conflict of interest

Authors AF and SV were employed by Petróleo Brasileiro S.A. - Petrobras.

The remaining author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fceng.2025.1687048/full#supplementary-material

References

Abduvayt, P., Arihara, N., Manabe, R., and Ikeda, K. (2003). Experimental and modeling studies for gas-liquid two-phase flow at high pressure conditions. *J. Jpn. Petroleum Inst.* 46, 111–125. doi:10.1627/jpi.46.111

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). "Optuna: a next-generation hyperparameter optimization framework," in *The 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2623–2631.

Akpan, I. B. (1980). *Two-phase metering. Master's thesis*. Tulsa, Oklahoma: University of Tulsa.

Al-Ruhaimani, F., Pereyra, E., Sarica, C., Al-Safran, E., and Torres, C. (2017). Experimental analysis and model evaluation of high-liquid-viscosity two-phase upward vertical pipe flow. *SPE J.* 22, 712–735. doi:10.2118/184401-PA

Alakbari, F. S., Ayoub, M. A., Awad, M. A., Ganat, T., Mohyaldinn, M. E., and Mahmood, S. M. (2025). A robust pressure drop prediction model in vertical multiphase flow: a machine learning approach. *Sci. Rep.* 15, 13420. doi:10.1038/s41598-025-96371-2

Alhashem, M. (2019). "Supervised machine learning in predicting multiphase flow regimes in horizontal pipes," in *Abu Dhabi international petroleum exhibition & conference* (Richardson, TX: Society of Petroleum Engineers).

Almabrok, A. A. (2013). *Investigation of the effects of high viscosity oil on flow patterns and pressure gradient in horizontal two-phase flow*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Alsaadi, Y. (2013). *Liquid loading in highly deviated Wells*. (master's thesis). The University of Tulsa, Tulsa, OK, United States.

Alves, I. N., Caetano, E. F., Minami, K., and Shoham, O. (1991). Modeling annular flow behavior for gas wells. *SPE Prod. Eng.* 6, 435–440. doi:10.2118/20384-PA

Andritsos, N. (1986). *Effect of pipe diameter and liquid viscosity on horizontal stratified two-phase flow*. (Ph.D. thesis). University of Illinois at Urbana-Champaign, Urbana, IL, United States.

Ansari, M. R., and Azadi, R. (2016). Effect of diameter and axial location on upward gas–liquid two-phase flow patterns in intermediate-scale vertical tubes. *Ann. Nucl. Energy* 94, 530–540. doi:10.1016/j.anucene.2016.04.020

Armand, A. A. (1946). The resistance during the movement of a two-phase system in horizontal pipes. *Izv. VTI* 1, 16–23.

Attia, M., Abdulraheem, A., and Mahmoud, M. A. (2015). "Pressure drop due to multiphase flow using four artificial intelligence methods," in *SPE North Africa technical conference and exhibition* (Richardson, TX: Society of Petroleum Engineers).

Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv Preprint*. [Preprint]. doi:10.48550/arXiv.1607.06450

Bakarji, J., Callaham, J., Brunton, S. L., and Kutz, J. N. (2022). Dimensionally consistent learning with buckingham pi. *Nat. Comput. Sci.* 2, 834–844. doi:10.1038/s43588-022-00355-5

Bardsley, J. M., Solonen, A., Haario, H., and Laine, M. (2014). Randomize-then-optimize: a method for sampling from posterior distributions in nonlinear inverse problems. *SIAM J. Sci. Comput.* 36, A1895–A1910. doi:10.1137/140964023

Barnea, D. (1987). A unified model for predicting flow-pattern transitions for the whole range of pipe inclinations. *Int. J. Multiph. Flow* 13, 1–12. doi:10.1016/0301-9322(87)90002-4

Barnea, D., Shoham, O., and Taitel, Y. (1982). Flow pattern transition for vertical downward two phase flow. *Chem. Eng. Sci.* 37, 5, 741–744. doi:10.1016/0009-2509(82)85034-3

Barnea, D., Shoham, O., Taitel, Y., and Dukler, A. E. (1985). Gas-liquid flow in inclined tubes: flow pattern transitions for upward flow. *Chem. Eng. Sci.* 40 (1), 131–136. doi:10.1016/0009-2509(85)85053-3

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. (2018). Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* 18, 1–43. [Preprint]. doi:10.48550/arXiv.1502.05767

Beggs, H. D. (1972). *An experimental study of two-phase flow in inclined pipes*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Beggs, D., and Brill, J. (1973). A study of two-phase flow in inclined pipes. *J. Petroleum Technol.* 25, 607–617. doi:10.2118/4007-PA

Brill, J., and Arirachakaran, S. (1992). State of the art in multiphase flow. *J. Petroleum Technol.* 44, 538–541. doi:10.2118/23835-PA

Brito, R. (2012). *Effect of medium oil viscosity on two-phase oil-gas flow behavior in horizontal pipes*. (Ph.D. thesis). University of Tulsa, Tulsa, OK, United States.

Brito, R., Pereyra, E., and Sarica, C. (2014). "Experimental study to characterize slug flow for medium oil viscosities in horizontal pipes," in *North American conference on multiphase technology of North American conference on multiphase production technology, BHR–2014–G4*, 9.

Brunton, S. L., Noack, B. R., and Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* 52, 477–508. doi:10.1146/annurev-fluid-010719-060214

Caetano, E. F. (1985). *Upward vertical two-phase flow through an annulus*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

[Dataset] Castro, M. S., Faller, A. C., and Vieira, S. C. (2025a). "Neural hydra. Certificate of registration for computer program no. BR512025002969-1, Brazilian national institute of industrial property (INPI)," in *Certificate issued on 2025-07-15. Valid for 50 years from 2025-01-01. Programming language: python*.

[Dataset] Castro, M. S., Vieira, S. C., and Faller, A. C. (2025b). "Método para simulação de escoamento multifásico por diferenciação automática e mídia de armazenamento legível por computador," in *Patent application no. BR102025012887-0, Brazilian national institute of industrial property (INPI)*.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). "Neural ordinary differential equations,". *Advances in neural information processing systems*. Editors S. Bengio, H. Wallach, H. Larochelle,

K. Grauman, N. Cesa-Bianchi, and R. Garnett (Red Hook, NY: Curran Associates, Inc.), 31.

Chen, X. T., Cal, X. D., and Brill, J. P. (1997). Gas-liquid stratified-wavy flow in horizontal pipelines. *J. Energy Resour. Technol.* 119(4), 209–216. doi:10.1115/1.2794992

Cheremisinoff, N. P. (1977). *An experimental and theoretical investigation of horizontal stratified and annular two-phase flow with heat transfer*. Potsdam, NY: Clarkson University.

Chisholm, D. (1983). *Two-phase flow in pipelines and heat exchangers*. Harlow, UK: Longman.

Churchill, S. W. (1977). Friction-factor equation spans all fluid-flow regimes. *Chem. Eng.* 84, 91–92.

Colebrook, C. F. (1939). Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws. *J. Institution Civ. Eng.* 11, 133–156. doi:10.1680/ijoti.1939.13150

Crowley, C. J., Sam, R. G., and Rothe, P. H. (1986). *Investigation of two-phase flow in horizontal and inclined pipes at large pipe size and high gas density (creare incorporated)*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* 12, 2121–2159. http://jmlr.org/papers/v12/duchi11a.html.

Eaton, B. A. (1966). *The prediction of flow patterns, liquid holdup and pressure losses occurring during continuous two-phase flow in horizontal pipelines*. (Ph.D. thesis). University of Texas at Austin, Austin, TX, United States.

Ezzatabadipour, M., Singh, P., Robinson, M., Guillen, P., and Torres, C. (2017). "Deep learning as a tool to predict flow patterns in two-phase flow," in *Proceedings of the workshop on data mining for oil and gas*.

Fan, Y. (2005). *An investigation of low liquid loading GasLiquid stratified flow in near-horizontal pipes*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Fedus, W., Zoph, B., and Shazeer, N. (2022). Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.* 23, 1–39. [Preprint]. doi:10.48550/arXiv.2101.03961

Felizola, H. (1992). *Slug flow in extended reach directional wells*. (master's thesis). The University of Tulsa, Tulsa, OK, United States.

Fox, R., McDonald, A., and Pritchard, P. (1985). *Introduction to fluid mechanics*, 7. New York: John Wiley & Sons.

Gawas, K. (2005). *Studies in low-liquid loading in gas/oil/water three phase flow in horizontal and near-horizontal pipes*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Gokcal, B. (2005). *Effects of high oil viscosity on two-phase oil-gas flow behavior in horizontal pipes*. (master's thesis). The University of Tulsa, Tulsa, OK, United States.

Gokcal, B. (2008). *An experimental and theoretical investigation of slug flow for high oil viscosity in horizontal pipes*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Gomez, L. E., Shoham, O., Schmidt, Z., Chokshi, R. N., and Northug, T. (2000). Unified mechanistic model for steady-state two-phase flow: horizontal to vertical upward flow. *SPE J.* 5, 339–350. doi:10.2118/65705-PA

Guillen-Rondon, P., Robinson, M. D., Torres, C., and Pereya, E. (2018). Support vector machine application for multiphase flow pattern prediction. *arXiv Preprint arXiv: 1806.05054*. doi:10.48550/arXiv.1806.05054

Guner, M. (2012). *Liquid loading of gas Wells with deviations from 0° to 45°*. (master's thesis). The University of Tulsa, Tulsa, OK, United States.

Hanafizadeh, P., Saidi, M. H., Nouri Gheimasi, A., and Ghanbarzadeh, S. (2011). Experimental investigation of air–water, two-phase flow regimes in vertical mini pipe. *Sci. Iran.* 18 (4), 923–929. doi:10.1016/j.scient.2011.07.003

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *2016 IEEE conference on computer vision and pattern recognition (CVPR)*, 770–778. doi:10.1109/CVPR.2016.90

Ioffe, S., and Szegedy, C. (2015). "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning (PMLR)*, 448–456.

Ishii, M., and Hibiki, T. (2006). *Thermo-fluid dynamics of two-phase flow*. New York, NY: Springer.

Johnson, N. J. B. A. G. W., Bertelsen, A. F., and Nossen, J. (2009). An experimental investigation of roll waves in high pressure two-phase inclined pipe flow. *Int. Journal Multiphase Flow* 35, 924–932. doi:10.1016/j.ijmultiphaseflow.2009.06.003

Kanin, E., Osiptsov, A., Vainshtein, A., and Burnaev, E. (2019). A predictive model for steady-state multiphase pipe flow: machine learning on lab data. *J. Petroleum Sci. Eng.* 180, 727–746. doi:10.1016/j.petrol.2019.05.055

Karami, H. (2015). *Three-phase low liquid loading flow and effects of MEG on flow behavior*. Ph.D. thesis.

Khaledi, H. A., Smith, I. E., Unander, T. E., and Nossen, J. (2014). Investigation of two-phase flow pattern, liquid holdup and pressure drop in viscous oil–gas flow. *Int. J. Multiph. Flow* 67, 37–51. doi:10.1016/j.ijmultiphaseflow.2014.07.006

Khan, M. S., Barooah, A., Lal, B., and Rahman, M. (2023). in *Multiphase flow systems and potential of machine learning approaches in cutting transport and liquid loading scenarios*, 27–57. doi:10.1007/978-3-031-24231-1_3

Kokal, S. L. (1987). *An experimental study of two phase flow in inclined pipes*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Kokal, S., and Stanislav, J. (1989a). An experimental study of two-phase flow in slightly inclined pipes—i. flow patterns. *Chem. Eng. Sci.* 44, 665–679. doi:10.1016/0009-2509(89)85042-0

Kokal, S., and Stanislav, J. (1989b). An experimental study of two-phase flow in slightly inclined pipes—ii. liquid holdup and pressure drop. *Chem. Eng. Sci.* 44, 681–693. doi:10.1016/0009-2509(89)85043-2

Kong, Q., Siauw, T., and Bayen, A. M. (2021). "Chapter 23 - boundary-value problems for ordinary differential equations (odes)," in *Python programming and numerical methods*. Editors Q. Kong, T. Siauw, and A. M. Bayen (Academic Press), 399–414. doi:10.1016/B978-0-12-819549-9.00033-6

Kouba, G. E. (1986). *Horizontal slug flow modeling and metering*. Ph.D. thesis.

Kristiansen, O. (2004). *Experiments on the transition from stratified to slug flow in multiphase pipe flow*.

Lai, Z., Mylonas, C., Nagarajaiah, S., and Chatzi, E. (2021). Structural identification with physics-informed neural ordinary differential equations. *J. Sound Vib.* 508, 116196. doi:10.1016/j.jsv.2021.116196

Lockhart, R. W., and Martinelli, R. C. (1949). Proposed correlation of data for isothermal two-phase, two-component flow in pipes. *Chem. Eng. Prog.* 45, 39–48.

Loshchilov, I., and Hutter, F. (2016). *Sgdr: stochastic gradient descent with warm restarts*. doi:10.48550/arXiv.1608.03983

Loshchilov, I., and Hutter, F. (2019). "Decoupled weight decay regularization," in *International conference on learning representations*.

Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E. (2021). Deepxde: a deep learning library for solving differential equations. *SIAM Rev.* 63, 208–228. doi:10.1137/19m1274067

Magrini, K. L. (2009). Liquid entrainment in annular gas-liquid flow in inclined pipes. (master's thesis). The University of Tulsa, Tulsa, OK, United States.

Marcano, R. (1996). *Slug characteritics for two-phase horizontal flow*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Meng, W. (1999). *Low liquid loading gas-liquid two-phase flow in near-horizontal pipes*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Minami, K. (1983). *Liquid holdup in wet-gas pipelines*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Mukherjee, H. (1979). An experimental study of inclined two-phase flow. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Nowlan, S., and Hinton, G. E. (1991). "Evaluation of adaptive mixtures of competing experts,". *Advances in neural information processing systems*. Editors R. Lippmann, J. Moody, and D. Touretzky (San Mateo, CA: Morgan-Kaufmann), 3.

Ohnuki, A., and Akimoto, H. (2000). Experimental study on transition of flow pattern and phase distribution in upward air–water two-phase flow along a large vertical pipe. *Int. Journal Multiphase Flow* 26, 367–386. doi:10.1016/s0301-9322(99)00024-5

Oliver, D., He, N., and Reynolds, A. (1996). in *Conditioning permeability fields to pressure data*. doi:10.3997/2214-4609.201406884

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). "Pytorch: an imperative style, high-performance deep learning library,". *Advances in neural information processing systems*. Editors H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Red Hook, NY: Curran Associates, Inc.), 32.

Pereyra, E., Torres, C., Mohan, R., Gomez, L., Kouba, G., and Shoham, O. (2012). A methodology and database to quantify the confidence level of methods for gas–liquid two-phase flow pattern prediction. *Chem. Eng. Res. Des.* 90, 507–513. doi:10.1016/j.cherd.2011.08.009

Pietrzak, M., and Płaczek, M. (2019). Void fraction predictive methods in two-phase flow across a small diameter channel. *Int. J. Multiph. Flow* 121, 103115. doi:10.1016/j.ijmultiphaseflow.2019.103115

Raissi, M., Perdikaris, P., and Karniadakis, G. (2019). Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707. doi:10.1016/j.jcp.2018.10.045

Roumazeilles, P. (1994). *An experimental study of downward slug flow in inclined pipes*. (master's thesis). The University of Tulsa, Tulsa, OK, United States.

Ruiz-Diaz, C. M., Hernández-Cely, M. M., and Estrada, O. A. G. (2021). A predictive model for the identification of the volume fraction in two-phase flow. *Cienc. Desarro.* 12, 49–55. doi:10.19053/01217488.v12.n2.2021.13417

Schmidt, Z. (1976). *Experimental study of gas-liquid flow in a pipeline-rise pipe system*. (master's thesis). The University of Tulsa, Tulsa, OK, United States.

Schmidt, Z. (1977). *Experimental study of gas-liquid flow in a pipeline-rise pipe system*. (master's thesis). The University of Tulsa, Tulsa, OK, United States.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., et al. (2017). "Outrageously large neural networks: the sparsely-gated mixture-of-experts layer," in *International conference on learning representations*.

Shippen, M., and Bailey, W. (2012). Steady-state multiphase flow - past, present, and future, with a perspective on flow assurance. *Energy & Fuels* 26, 4145–4157. doi:10.1021/ef300301s

Shmueli, A., Unander, T., and Nydal, O. (2015). "Characteristics of gas/water/viscous oil in stratified-annular horizontal pipe flows," in *Offshore technology conference Brasil (OTC)*.D021S024R004

Shoham, O. (2006). *Mechanistic modeling of gas-liquid two-phase flow in pipes*. Richardson, TX: Society of Petroleum Engineers. doi:10.2118/9781555631079

Swamee, P. K., and Jain, A. K. (1976). Explicit equations for pipe-flow problems. *J. Hydraulics Div.* 102, 657–664. doi:10.1061/jyceaj.0004542

Taitel, Y., and Barnea, D. (1990). Two-phase slug flow (Elsevier). *Of* Adv. in Heat Transf. 20, 83–132. doi:10.1016/S0065-2717(08)70026-1

Taitel, Y., and Dukler, A. E. (1976). A model for predicting flow regime transitions in horizontal and near horizontal gas-liquid flow. *AIChE J.* 22, 47–55. doi:10.1002/aic.690220105

Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*, xii. doi:10.1137/1.9780898717921

Usui, K., and Sato, K. (1989). Vertically downward two-phase flow, (i) void distribution and average void fraction. *J. Nucl. Sci. Technol.* 26, 670–680. doi:10.1080/18811248.1989.9734366

van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-sne. *J. Mach. Learn. Res.* 9, 2579–2605. Available online at: http://jmlr.org/papers/v9/vandermaaten08a.html

Viana, F. (2017). *Liquid entrainment in gas at high-pressures*. Ph.D. thesis.

Vongvuthipornchai, S. (1982). *Experimental study of pressure wave propagation in two-phase mixtures*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Vuong, D. H. (2016). *Pressure effects on two-phase oil-gas low-liquid-loading flow in horizontal pipes*. Ph.D. thesis.

Werbos, P. (1974). *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. Cambridge: Harvard University. Ph.d. thesis.

Yamaguchi, K., and Yamazaki, Y. (1984). Combinated flow pattern map for cocurrent and countercurrent air-water flows in vertical tube. *J. Nucl. Sci. Technol.* 21, 321–327. doi:10.1080/18811248.1984.9731053

Yang, J. S. (1996). *A study of intermittent flow in downward inclined pipes*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Yuan, G. (2011). *Liquid loading of gas wells*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Zheng, G. (1989). *Two-phase slug flow in upward pipes*. (Ph.D. thesis). The University of Tulsa, Tulsa, OK, United States.

Zhu, L.-T., Chen, X., Ouyang, B., Yan, W.-C., Lei, H., Chen, Z., et al. (2022). Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors. *Industrial & Eng. Chem. Res.* 61, 9901–9949. doi:10.1021/acs.iecr.2c01036

Zhu, J., Chen, X., He, K., LeCun, Y., and Liu, Z. (2025). "Transformers without normalization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.