



Enhancing vortex-flow-meter precision using physics-informed contrastive learning

Stephan Wernli^{a,b}*, Marc Hollmach^b, Christian Franzmann^b, Daniel Kessler^b, Henrique Fernandes^{a,c}, Lilach Goren Huber^d

^a Faculty of Engineering and Applied Sciences, Cranfield University, United Kingdom

^b Endress+Hauser Flowtec AG, Switzerland

^c Faculty of Computing, Federal University of Uberlandia, Brazil

^d School of Engineering, Zurich University of Applied Sciences, Switzerland

ARTICLE INFO

Keywords:

Sensor calibration correction
Physics-informed deep learning
CFD simulation
Contrastive regression
Deep learning
Vortex flow meters

ABSTRACT

High precision vortex-based flow measurement devices are subject to systematic measurement errors caused by installation effects and complex flow conditions that cannot always be directly measured or compensated. Correcting such systematic measurement errors is crucial for achieving a yet higher measurement accuracy and reliability. In this work we introduce a hybrid framework for error correction of vortex flow meters. The method uses physically-engineered features derived from computational fluid dynamics (CFD) simulations as inputs to a deep contrastive regression neural network. The network learns latent representations that are useful for predicting measurement errors under various pipe geometries and flow regimes. We demonstrate the effectiveness of this representation learning by testing the prediction performance of the framework under new pipe geometries not seen during training. The method demonstrates the potential of advanced deep learning models to extract physically meaningful features for error prediction tasks in complex, highly non-linear flow setups, in which CFD simulations reach their computational limits.

1. Introduction

Volumetric flow measurement devices are critical components in the process industries, essential for applications such as tracking oil and gas transfers, monitoring emissions, and ensuring efficient production [1]. To provide accurate measurements of the flow velocity, the installed flow meter is calibrated using predefined calibration protocols. In the case of vortex-based flow meters, these protocols are derived based on the assumption of a fully developed flow profile, typically observed in a straight pipe far away from any obstacles or flow disturbances [2,3].

In real-world industrial applications, the actual flow profile in pipe installations rarely corresponds to a fully developed flow profile. Instead, common pipe fittings include elbows, valves, pumps, or changes in pipe diameter that distort the velocity profile. Therefore, for the vast majority of industrial installations ideal installation conditions are rarely achievable due to economic and spatial constraints, thus non-ideal conditions are standard. As a result of these distortions, the sensor readings under standard calibration procedures often suffer biases and systematic errors [1,2], carrying profound implications for industries that rely on precise flow measurements for their operational efficiency, safety and compliance with regulatory standards.

In the case of vortex flow meters, the two optional solutions to compensate for systematic measurement errors have proven inadequate: on one hand, an in-situ calibration of every new installation is too labor-intensive and impractical [4]. On the other hand, estimating the errors based on Computational Fluid Dynamic (CFD) flow simulations for specific setups [1,5] is infeasible for vortex flow meters. The reason is that the flow meter itself introduces unsteady flow phenomena, making the simulation accuracy and computational times impractical for commercial use.

In this paper we propose a novel approach to correcting systematic measurement errors of vortex flow meters, caused by non-ideal pipe fittings. The method combines data measured under several non-ideal conditions together with partial CFD simulations of the flow profile in the absence of the flow meter (and thus under steady flow conditions that allow for efficient and reliable CFD outcomes). A physics-informed deep learning algorithm uses both of these inputs to predict a calibration factor for a new installation, not seen during the model training. In particular, we demonstrate a first application of a deep contrastive regression network in order to extract physically meaningful features

* Corresponding author at: Faculty of Engineering and Applied Sciences, Cranfield University, United Kingdom.

E-mail address: Stephan.Wernli@cranfield.ac.uk (S. Wernli).

<https://doi.org/10.1016/j.flowmeasinst.2026.103207>

Received 27 October 2025; Received in revised form 26 December 2025; Accepted 13 January 2026

Available online 3 February 2026

0955-5986/© 2026 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

for error prediction tasks in complex, highly non-linear flow setups, in which CFD simulations reach their computational limits.

2. Related work

Recent advances in machine learning (ML) and physics-informed modeling have significantly impacted sensor calibration and error correction [6], particularly in flow measurement and fluid dynamics [7]. However, a critical research gap exists at the intersection of two domains: (1) sensor calibration using ML and physics-informed approaches, (2) hybrid modeling techniques for fluid dynamics problems. This gap becomes particularly pronounced for vortex flow sensors, which pose unique challenges due to their inherent computational complexity and the need for generalization to new installations. In the following, we review the published work in these two domains, highlighting the contribution of the present paper.

Sensor calibration and error correction. The application of ML techniques for sensor calibration and error correction has gained significant traction across various domains. General sensor calibration has seen successful applications of supervised ML models, particularly effective in scenarios where sensor drift, environmental noise, or manufacturing variability degrade accuracy [8–10]. These approaches demonstrate that even low-cost sensors can achieve near-reference accuracy when calibrated with robust ML techniques, commonly applied in air quality monitoring, temperature and humidity sensing, and portable gas detection systems [11]. For flow measurements in particular, ML and DL models, such as LSTM and CNN, have been applied to compensate for drift and errors [9,12–15]. The clear disadvantage of ML methods is their poor ability to extrapolate and generalize to predictions beyond the training conditions. This limitation is particularly problematic for industrial applications, where sensors must perform accurately across diverse and previously unseen operational environments. Thus, pure ML methods are often ineffective for learning the calibration parameters in new installations, away from the training conditions. An alternative to ML methods is using physics-based models for flow meter error correction. This approach is proposed in a recent paper employing CFD simulations to calculate the effect of non-ideal pipe fittings in ultrasonic flow meters on flow measurement errors [16]. Unlike ultrasonic meters, where full CFD-based approaches are feasible, vortex meters require about a hundred times longer simulation times due to their inherent flow unsteadiness. Due to these prohibitive computational times, the approach proposed in [16] is not applicable for vortex flow meters in operational settings.

Combining physics with ML approaches using hybrid methods is a growing research field. A recent work suggests combining ML with physical models for efficient calibration of engine mass flow estimates [17]. The proposed approach is, however, limited to using measured data from the installed system, and does not address the calibration of an unmeasured setup. In contrast, here we propose a hybrid approach to directly address the common operational scenario in which a new flow meter is installed and must be calibrated based on data collected previously from other measurement setups. This corresponds to an out-of-distribution generalization of the calibration algorithm, which is achieved by exploiting physical CFD simulations to compensate for the absence of measured data. Extracting features from CFD simulated flow fields as input to DL models allows us to avoid a full simulation of the flow meter which would be infeasible for vortex flow meters.

We note that hybrid approaches combining CFD with ML methods (e.g. Reduced Order Models or neural network surrogates) are often used to accelerate CFD simulations for predicting the flow field. The measurement calibration task that we address in this work is fundamentally different: the full simulation model in the presence of the vortex meter is computationally infeasible and thus cannot be directly replaced by a surrogate model. Instead, our model learns features from

a simpler CFD flow field, and combines them with measured data to bridge the simulation gap, and not only accelerate the simulation. This approach allows for calibration in a new setup, for which measured data is unavailable.

Deep learning (DL) for feature extraction in fluid dynamics. DL applications in fluid dynamics have proliferated extensively, with convolutional neural networks demonstrating remarkable effectiveness for feature extraction from CFD flow fields [18,19]. The ability of DL models to extract useful representations for complex fluid dynamic tasks has inspired numerous applications. The most common applications are for dynamic modeling of flow properties [20–22] and accelerated equation solving and simulation using convolutional (CNN) or physics-informed neural network (PINN) architectures [23–29]. Our proposed approach is inspired by the previous success of CNN architectures to extract meaningful features from CFD flow fields. However, the application of the powerful feature extraction capabilities specifically for sensor error correction remains largely unexplored in the scientific literature, and is one of the main contributions of our work.

To achieve a powerful feature extraction out of the CFD flow fields, with generalization capabilities beyond the training set, we apply a contrastive learning neural network [30]. Contrastive learning has gained popularity in recent years for a multitude of classification tasks [31]. For regression tasks as the one we target here, an adaptation of the concept has been proposed [32], and applied to gaze direction and age estimations from human images [33]. Further applications of contrastive regression include action quality assessments from videos [34,35], speech quality assessment from audio signals [36], forest inventory mapping from satellite images [37], and remaining useful life prediction from sensor data [38]. However, to the best of our knowledge there has been no application of contrastive regression approaches to technical images, and in particular to CFD flow fields. Our application of contrastive regression to the prediction of calibration errors from CFD images is a crucial modeling ingredient that enables generalization of the error prediction to unseen setups.

Summary of the paper contributions. The comprehensive review of existing literature reveals several critical gaps, in the application as well as the methodology, that are covered by the present paper.

- **Error correction for vortex flow sensors:** To the best of our knowledge, no existing research addresses measurement error correction for vortex flow sensors using a hybrid approach based on combining measured-data with CFD simulations.
- **Out-of-distribution generalization:** Current approaches for sensor error correction fail to provide reliable error correction for new installations or operational conditions not represented in training data. Our method enables out-of-distribution generalization, using contrastive regression to extract predictive features that are robust to variations in the pipe geometry.
- **Computational feasibility:** Existing methods for flow measurement error correction either rely on prohibitively complex CFD simulations or suffer from poor generalization capabilities when using pure ML approaches. Our proposed approach overcomes both challenges.
- **Physics-informed ML for Error Correction:** While hybrid methods excel in fluid dynamics simulation, their application to sensor error correction, and particularly for complex flow sensing systems, is presented here for the first time.
- **Contrastive regression for fluid dynamic applications:** Our work is a first demonstration of using contrastive regression to optimize deep representation learning from CFD simulation outputs.

Our work addresses these gaps by proposing the first hybrid approach for vortex flow sensor error correction that combines

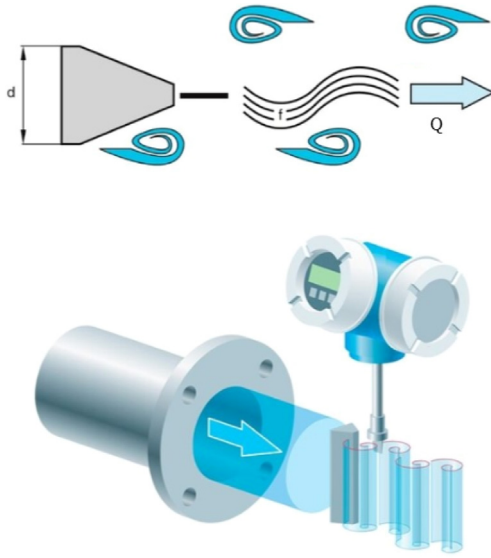


Fig. 1. The measuring principle of vortex flow meters based on the von Kármán vortex street phenomenon. When fluid flows past the bluff body (diameter d), alternating vortices are shed at a frequency f that is directly proportional to the flow rate Q . The proportionality factor c must be estimated for every new installation. (Figure adapted from [40])

the feature extraction capabilities of deep learning with physics-informed constraints to enable accurate out-of-distribution predictions using realistic computational resources.

This represents a significant advancement in both sensor calibration methodology and practical industrial application, enabling reliable error correction for new installations without requiring extensive on-site calibration or complex simulation tools. Moreover, the methodology demonstrates a first successful application of deep contrastive regression to fluid dynamic problems.

3. Description of the system

The measurement principle of vortex flow meters is based on the von Kármán vortex street phenomenon [39]. The flow meter consists of a bluff body which is immersed in the flowing medium. Vortices are generated in a consistent pattern downstream (see Fig. 1), and the frequency f at which these vortices are shed is directly proportional to the volumetric flow rate Q in the pipe,

$$f = c_{\text{ref}} Q. \quad (1)$$

Thus, by measuring the shedding frequency, the flow rate can be calculated with a high degree of precision, assuming that the proportionality constant c_{ref} is known. This constant, also known as the calibration factor (calf), can be accurately determined in a standard calibration process under ideal, fully developed straight pipe flow conditions.

In real-world installation scenarios, however, flow meters are often installed close to pipe bends or obstructions, and thus the flow conditions deviate significantly from the ideal assumptions. Under such non-ideal circumstances, past observations indicate a potential deviation of up to 4.5% of the true calibration factor c from the ideal one c_{ref} . In the following we describe our proposed method for the prediction of the true calibration factor for a new non-ideal installation. The true calibration factor is termed calf and is defined as

$$c = \frac{f}{Q}. \quad (2)$$

4. Method overview

The proposed method is aimed at predicting c for a new installation, using the measured values in a set of non-ideal flow installations. To this end, we propose to train a deep neural network with two data types: (i) the measured calibration factor c for a set of non-ideal flow installations, and (ii) the steady flow field in the absence of the flow meter, derived for each installed setup using CFD simulations. The paired data, containing the flow fields and their c labels is used during training.

At inference, however, we assume a standard situation for a newly installed flow meter, in which the steady CFD flow field can be obtained and the calibration factor c cannot be measured, and must be predicted by the model.

To achieve this, we train a deep representation learning model. This is a deep neural network, that during training receives CFD flow fields as inputs, along with their corresponding calibration factor labels for several non-ideal flow installations. The network learns informative features of the flow field that are useful for the calibration factor prediction. Since the calibration factor values are continuous scalars, the neural network performs a regression task. At inference, when the target calibration factor is not available, the learned representations are used to map the available CFD fields to a predicted calibration factor value (see Fig. 2 for a schematic overview).

In the following sub-sections we describe (i) the lab data of calibration factor measurements used as the regression target variable (ii) the CFD simulation images used as the model inputs, and (iii) the physics-informed deep representation learning model.

4.1. Dataset description

To predict the correct value of the calibration factor c for a newly installed vortex meter, we propose to rely on a small set of lab measurements. Since the new installation may differ in various parameters from the lab conditions, we train a model that learns to transfer relevant information from lab conditions to the new installation. To this end, we performed a set of lab measurements in which the calibration factor was extracted for 420 different configurations. In a second step, we split the configurations into training and test (or inference) configurations. The training configurations are used to train a model that learns to predict the calibration factor using two dimensional (2D) flow fields from physical CFD simulations. The test configurations are held out for testing the trained model.

The calibration factor depends on several parameters. On one hand parameters related to the properties of the fluid, such as its density (ρ), dynamic viscosity (μ) and flow velocity (u). In addition, since most real-world installations are of a complex geometry (e.g with multiple bends in the pipes) the calibration factor depends on the geometric properties of the installation. In particular, the distance of the sensor to the inlet, the shape of the pipe (how many bends and in which directions) and the orientation at which the sensor is installed in the pipe, with respect to the pipe bends.

In order to cover the important influencing factors we conducted a set of experiments, measuring the calibration factor while varying the following parameters:

1. The inlet disturbance (G). Three pipe geometries were measured (see Fig. 3): (i) a pipe with a single-bend referred to as “Single bend” (ii) a pipe with an in-plane double bend, referred to as “Double bend 2D” and (iii) a pipe with an out-of-plane double bend, referred to as “Double bend 3D”, as shown in Fig. 3. In both double bend configurations a straight pipe section with a length of 60 mm is placed between the two bends.

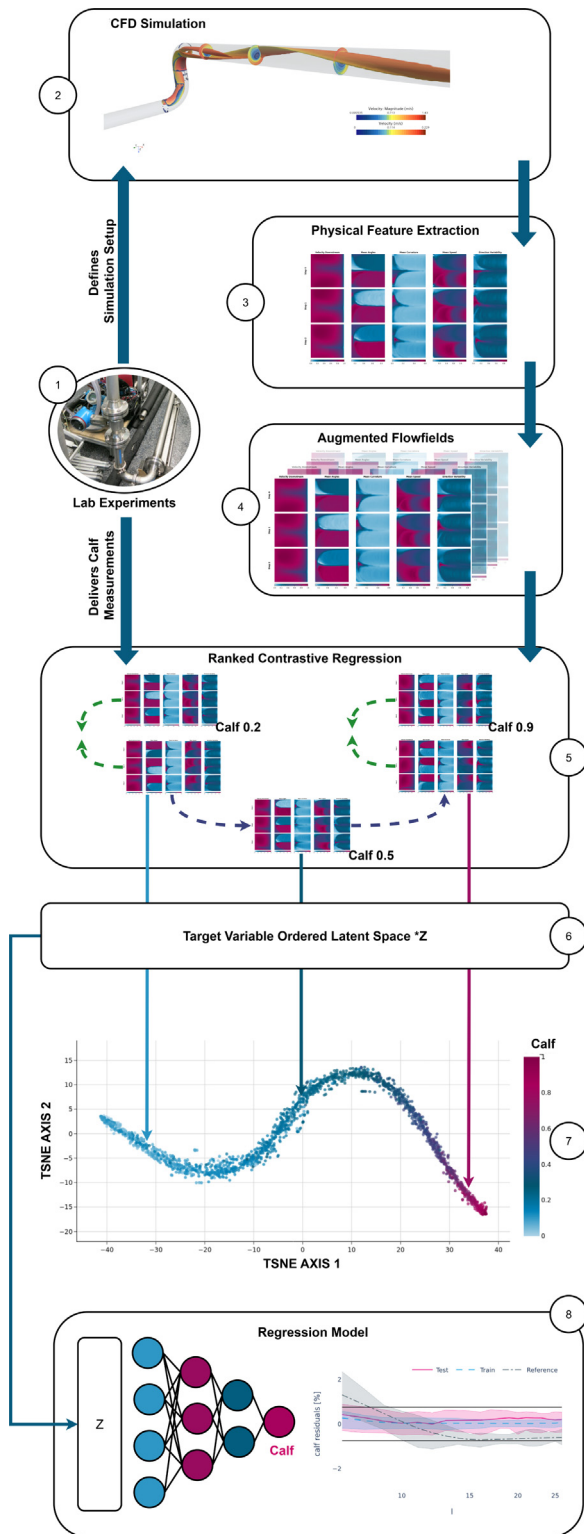


Fig. 2. Schematic overview of the measurement error correction approach. (1) experimental acquisition of calibration data (2) numerical CFD simulation (3) physical feature extraction (4) data augmentation (5) DL model training with a ranked contrastive loss (6) the resulting latent representation z (7) t-SNE visualization of z (8) regression model mapping of z onto the calibration factor (calf).

- The pipe diameter D . Despite the theoretical expectation that the pipe diameter should not affect the calf beyond its effect on the Reynolds number, it is observed in practice that non-ideal installations with different pipe diameters require different calibration factors. To allow for this variability in the model we perform experiments with two different diameters. The two pipe diameters correspond to NPS (Nominal Pipe Size) of $1\frac{1}{2}$ " (labeled as 'DN40' throughout this paper) and 2" (labeled as 'DN50') respectively, according to the ASME B36.10M standard. All bend radii are $1.32D$ for DN40 and $1.45D$ for DN50.
- The device orientation θ . Measurements were conducted with $\theta = 0^\circ$ and $\theta = 90^\circ$.
- The inlet distance L , measured from the last bend outlet to the bluff body of the sensor, normalized with the pipe diameter D : $l = L/D$. Measurements with $l = 7, 12, 17, 22, 27$ were conducted.
- The Reynolds number $Re = \rho u D / \mu$, capture the dependence on the fluid properties. The experiments were conducted at 7 Reynolds numbers ranging from 12k to 270k.

We obtained the values of the calibration factor in experiments covering 3 inlet disturbances, 2 diameters, 2 orientations, 7 Reynolds number values, and 5 inlet distances, amounting to 420 different configurations.

A thorough understanding of the influencing factors and their relative importance is still lacking, making it difficult to fully explain the complex dependencies of the calibration factor. To demonstrate the non-trivial nature of the calibration factor curve, Fig. 4 compares the measured calibration factor values of 4 setups, including single bend and double bend 3D, each with both pipe diameters of DN40 and DN50.

From Fig. 4 it is seen that no simple transformation can explain the complex variability of the calibration factor over the entire parameter space. In particular, transferring the calibration factor values between two different inlet disturbances or two different pipe diameters is non-trivial and cannot be done with simple heuristics.

Motivated by this challenge, the method we propose is aimed at exploiting the available physical information about the flow properties in order to extract latent features that are relevant for an accurate prediction of the calibration factor in complex installations. To this end we make use of CFD simulations of the flow field, that can be carried out with reasonable numerical effort for a new pipe installation as long as the vortex meter is omitted from the simulation. In the following section we elaborate on the outputs of the CFD simulation, which are used as inputs to a deep representation learning network, predicting the calibration factor.

4.2. CFD simulation

The input to the proposed physics-informed model are features extracted from CFD simulations. For each configuration with a lab measurement of the calibration factor, we computed the full flow field in the relevant inlet distances using StarCCM+ [41]. The flow field is simulated on a 3D grid. The discrete cross-sectional cuts of the flow field are enumerated using the normalized inlet distance $l = 1 \dots l_{\max}$, as illustrated in Fig. 3. Each of these cross-sectional cuts can be represented as a 2D grid of 3D-flow vectors describing the flow direction and intensity at each point on the simulated grid.

It is important to note that the CFD simulation computes the flow field in the *absence* of the flow meter, and thus cannot be used for a direct calculation of the calf. Including the flow meter's bluff body in the simulation would disruptively increase the complexity, resulting in a prohibitively long computational time. In fact, even the simplest simulation geometries, such as the ideal straight pipe, suffers from high inaccuracies when the flow meter is included in the model. To demonstrate this, Fig. 5 shows the calibration factor, as calculated from a CFD simulation of a straight pipe for different Reynolds numbers, when the flow meter is included in the simulation. It is seen that the results fluctuate strongly in the range of 2%, significantly above

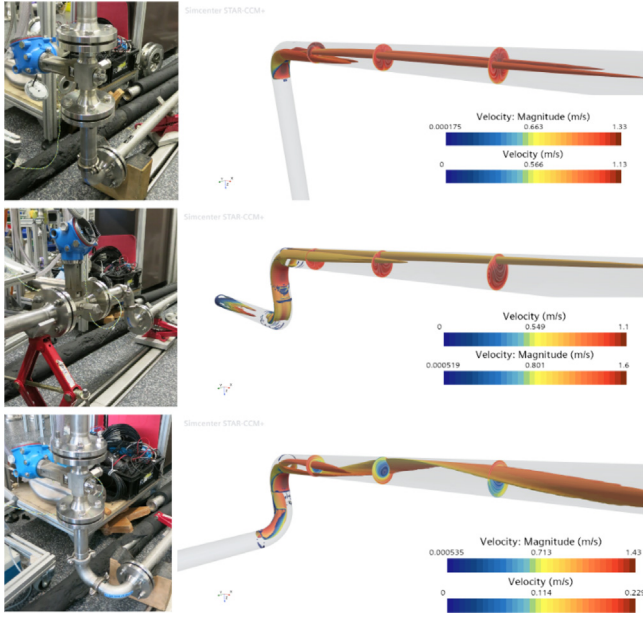


Fig. 3. Non-ideal inlet disturbances. The experimental setups of 3 inlet disturbances (left) and their corresponding CFD simulated flow fields (right). Top: Single bend, center: Double bend 2D, bottom: Double bend 3D.

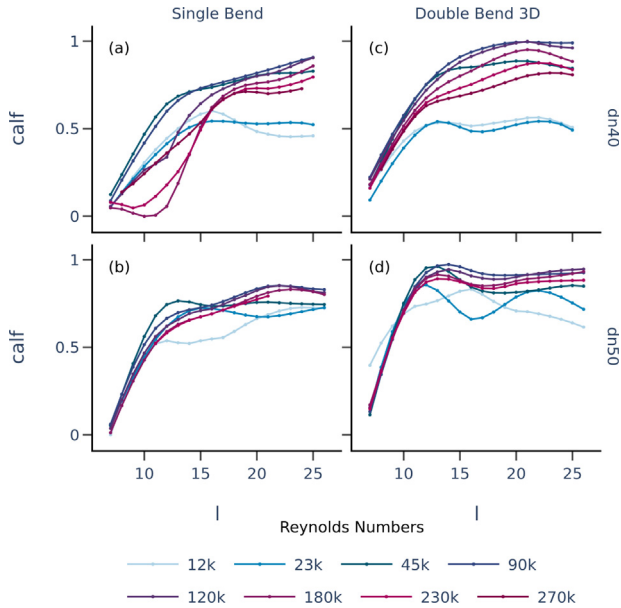


Fig. 4. Calibration factor variability. The measured (normalized) calf values are shown as function of the inlet distance l for 4 different setups: (a) Single bend, DN40 (b) Single bend DN50 (c) Double bend 3D, DN40 (d) Double bend 3D, DN50. Different Reynolds numbers are plotted in different colors.

the acceptable error range and are additionally dependent on the grid resolution. Including the bluff body in the simulation domain requires transient and scale-resolving simulations to accurately capture the vortex shedding dynamics. The difficulty of accurate simulations in this case stems from the trade-off between resolution and complexity needed to model turbulent features, while keeping the computational cost manageable.

To achieve accurate calibration factor estimates, yet taking advantage of existing CFD simulation capabilities, we propose to use the simulation when it is accurate and reliable, i.e. in the steady regime. To

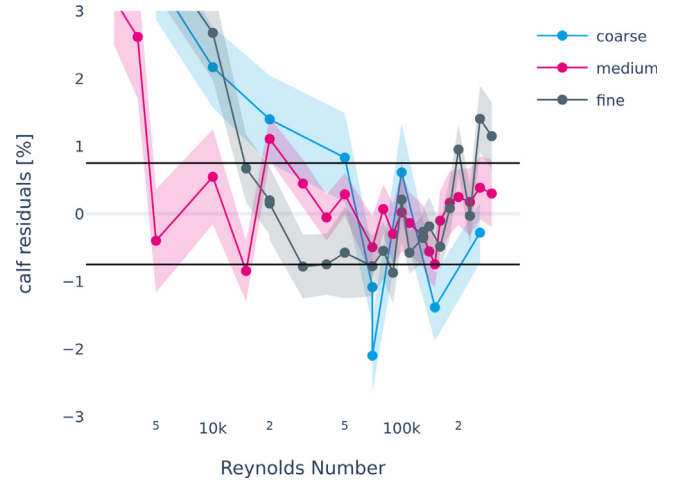


Fig. 5. Accuracy of CFD-based calibration factors. The calibration factors, calculated for a straight pipe using full CFD simulations (including the flow meter) with 3 different mesh resolutions, are shown as a function of the Reynolds number.

this end, our model receives simulation flow fields *in the absence* of the vortex meter, and combines them with measured data. The restriction to the steady regime allows the application of accurate state-of-the-art CFD and turbulence models while keeping the computational cost orders of magnitude below the transient simulations that include the bluff body.

Below we describe the steps of the method and demonstrate its ability to extract meaningful features from the flow field, despite the fact that the device itself and the vortices it sheds are not modeled.

4.3. Physics-informed contrastive regression for calibration factor prediction

As explained above, an estimate of the calibration factor for a new installation based on pure physical modeling is not feasible. A direct in-situ measurement of the calibration factor is also impractical in most cases. In this section we propose a hybrid model combining both physical simulation and lab measurement data to enable an accurate prediction of the calibration factor for a new installation. We consider a training set $\{\mathcal{U}_j, c_j\}_{j \in \mathcal{T}}$ for which both the measured calibration factor values c_j and the calculated CFD flow fields $\mathcal{U}_j$ are available for each configuration j . Each configuration in the data is defined by a distinct set of parameters $\{G_j, D_j, \theta_j, Re_j, l_j\}_{j \in \mathcal{T}}$.

Using the training configurations, we train a deep learning regression model $c_j = f(\mathcal{U}_j)$ to map the CFD flow field onto the corresponding calibration factor value. To predict the calibration factor value for a newly installed configuration $i \notin \mathcal{T}$, we use the trained model with the CFD simulation of the flow field $\mathcal{U}_i$ as input. Prior to the training we perform two steps of data augmentation and feature extraction that are described below.

4.3.1. Physics-informed data augmentation

To allow for proper training of deep learning models we increase the dataset size using three data augmentation steps:

1. **Interpolation of calibration factor values.** By imposing the continuity of the calibration factor value c as function of the inlet distance l , we introduce additional calibration factor values between the available measurements by applying Clough–Tocher triangular interpolation [42] to generate a continuous surface from the discrete measurement points. Importantly, the interpolation only generates values within the physically measured

bounds, and we validated the interpolated surfaces for continuity and consistency with the original measurement points to ensure physical plausibility. We then complement the calibration factor data by producing the simulated flow fields corresponding to each of the interpolated inlet distances. This step augments the number of samples from 420 to 5040.

2. *Symmetry-based augmentation.* In the context of vortex flow meters, the symmetry of the bluff body's geometry ensures that certain transformations, such as 180-degree rotations or mirroring, do not fundamentally alter the vortex shedding process, thus result in the same calibration factor value. Using the rotational and mirroring symmetries we obtain a further augmentation of the dataset from 5040 to 20160 samples.
3. *Gaussian noise augmentation.* To further enhance the robustness of the model and improve generalization to measurement noise, we introduce a noise-based data augmentation. Following best practices in ML-driven CFD modeling [43], we apply multiplicative Gaussian noise augmentation. For each original flow field, two noisy variants are generated according to: $\mathbf{u}_{\text{noisy}} = \mathbf{u} \cdot (1 + \mathcal{N}(0, \sigma^2))$ where $\sigma = \alpha \cdot \max(|\mathbf{u}|)$ is computed channel-wise, with the noise intensity α randomly sampled from the uniform distribution $\alpha \sim \mathcal{U}(0, 0.05)$. A sensitivity analysis was conducted to determine the optimal noise level, revealing that 0% noise provides baseline performance but with potential overfitting, and noise levels exceeding 10% lead to performance degradation due to excessive perturbations. Our selected threshold of 5% noise offers an optimal balance between robustness and accuracy. This augmentation strategy has been shown to enhance generalization in neural networks trained on flow field data. The multiplicative noise simulates relative uncertainties in CFD measurements while preserving the underlying physical characteristics of the flow fields. This augmentation increases the dataset size from 20160 to 60480 samples.

4.3.2. Physics-informed feature extraction

Using CFD we simulate the full flow field $\mathbf{U}_j$ for a given configuration j . This includes the longitudinal flow u along the pipe direction, and the secondary flow field along the pipe cross-section, described by the in-plane flow components v and w . From the simulated raw flow field components u, v, w , we calculate a set of 5 physical features for a given configuration. This feature generation is motivated by the physical understanding of the potential variables influencing the dynamics of the flow at the vicinity of the bluff body, thereby having possible effects on the measured flow. The features were selected from a larger set of candidate flow metrics, including velocity-based, vorticity-based, strain rate, streamline-based, and turbulence features. This feature selection captures flow phenomena relevant to vortex shedding and flow dynamics near the bluff body while minimizing feature redundancy. We note that optimal feature selection for physics-informed flow field representations remains an active area of research. Our future research will investigate alternative feature sets and automated feature selection approaches. The following 5 features are calculated for each configuration:

1. The downstream velocity field $u(x, y)$ at position (x, y) on the cross-section.
2. The mean speed in the transverse plane along a streamline,

$$\bar{s} = \frac{1}{N} \sum_{s=1}^N \sqrt{v(x_s, y_s)^2 + w(x_s, y_s)^2}.$$

where s is the position along a local streamline with N grid points.

3. The mean local in-plane direction of the stream,

$$\bar{\theta} = \frac{1}{N} \sum_{s=1}^N \theta_s,$$

where $\theta_s = \arctan(v(x_s, y_s), w(x_s, y_s))$.

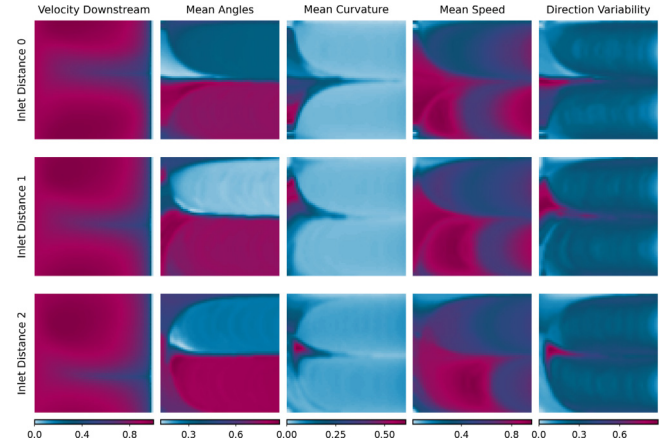


Fig. 6. Input features for a given configuration. Example of the $5 \times 3 \times 64 \times 64$ input feature maps for a “Double bend 3D” configuration (DN50, 0° orientation, $Re = 1.8 \cdot 10^5$). The rows represent three consecutive inlet distances ($l = 7, l = 6, l = 5$), providing the spatial sequence from which the model abstracts the dynamical properties and evolution of the flow field. Each column displays one of the five physical variables after subtraction of the ideal flow profile.

4. The mean local curvature of the streamlines,

$$\bar{\kappa} = \frac{1}{N} \sum_{s=1}^N \left| \frac{d\theta_s}{ds} \right|$$

5. The direction variability as a proxy to the local vorticity,

$$\sigma_\theta = \sqrt{\frac{1}{N} \sum_{s=1}^N (\theta_s - \bar{\theta})^2}$$

Each of the 5 variables can be represented as a 2D feature map, corresponding to the magnitude of the variable at each simulated grid point of the cross section at the corresponding inlet distance.

To capture the dynamical properties of the flow at inlet distance l_j , we incorporate the extracted flow variables from the preceding distances l_{j-1} and l_{j-2} in addition to the flow properties at l_j . The resulting input space $\mathbf{X}_j$ for sample j is therefore composed of 15 2D feature maps, where each of the five physical variables is calculated at three consecutive inlet distances.

As a final step of the feature calculation, we subtract from each 2D map the corresponding map calculated in ideal flow conditions (for a straight pipe). In this way, we aim to narrow down the feature learning to flow properties that are responsible for the calibration factor correction due to non-ideal flow. The cylindrical 2D cuts are mapped to a square polar grid ($n \times n$) using bilinear interpolation of the flow field values from Cartesian (Y, Z) to polar coordinates (r, θ). The resulting dimension of the input $\mathbf{X}_j$ is $5 \times 3 \times 64 \times 64$, consisting of 5 channels of $3 \times 64 \times 64$ 3D features.

Fig. 6 shows an example of the input features for one selected configuration, specified by “Double bend 3D” inlet disturbance with diameter DN50, 0° orientation, Reynolds number of $1.8 \cdot 10^5$ at inlet distance $l = 7$. In the figure the features are arranged as 5 columns of $3 \times 2D$ channels. Each row corresponds to a different inlet distance, allowing for a direct comparison of a certain physical property at different distances from the pipe inlet. The comparison shows an evolution of the input features, indicating the evolution of the flow field along the pipe. As explained below in Section 5.4, an ablation study confirms the intuitive expectation that input features at various inlet distances enable the model to capture dynamical properties of the flow, which are relevant for the prediction of the calibration factor.

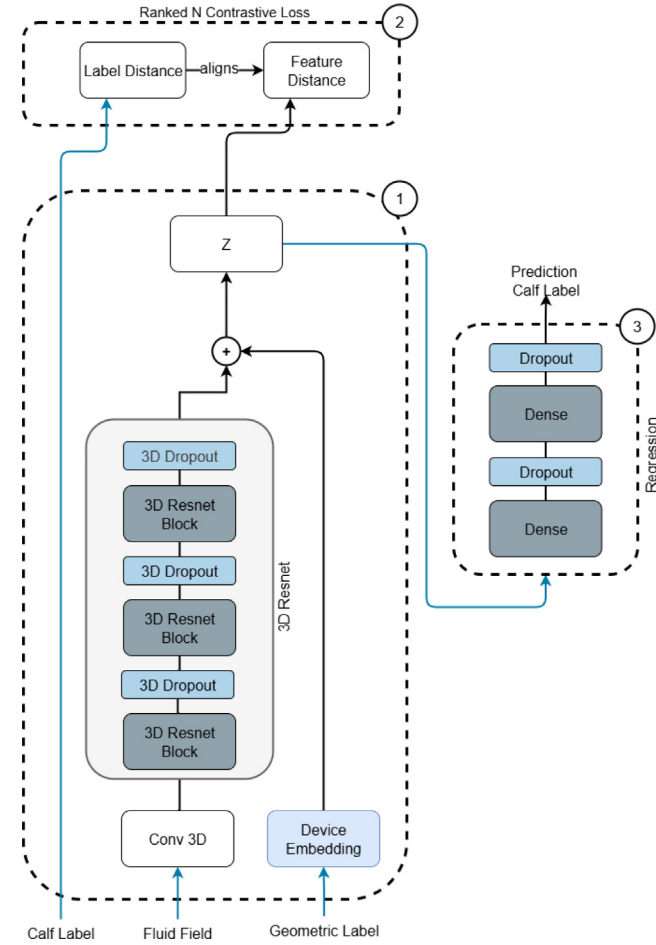


Fig. 7. Model architecture. The flow field features are fed into a convolutional feature extraction block (1) whose output is the latent representation z . The representation learning is trained using a Ranked Contrastive Loss (2). The resulting representation is used as input to a regression block (3) whose output is the predicted calf.

4.3.3. Deep contrastive regression for representation learning

We train a regression model to predict the calibration factor c_j of a given configuration j using the extracted flow features $\mathbf{X}_j$. The input $\mathbf{X}_j$ is fed into a 3D convolutional encoder neural network that outputs a latent representation z_j of the entire feature space. The latent representations are then fed into a contrastive regression module, whose role is to align the latent representations according to their calibration factor value. This is done by incorporating a ranked contrastive loss, proposed in [33] for a different application. The optimally aligned features z^* are then fed into a regression model to predict the calibration factor value. The model architecture is depicted in Fig. 7, and its main ingredients are explained in detail below.

The encoder network. The encoder is constructed of a hierarchical 3D CNN architecture to process volumetric CFD data, motivated by previous works demonstrating the effectiveness of residual architectures for fluid dynamics applications [44–48]. The network takes 5-channel 3D flow fields as input and produces a 32-dimensional latent representation through the following encoder pathway: (i) an initial 3D convolution ($5 \rightarrow 32$ channels, kernel size 3^3) with ELU activation and 3D dropout ($p = 0.2$), (ii) two residual blocks with progressive channel expansion ($32 \rightarrow 64 \rightarrow 96$) and spatial downsampling (factor $2 \times$ per block), each followed by 3D dropout ($p = 0.2$). Each residual block consists of two 3D convolutional layers with batch normalization, ELU activation, and skip connections. Global average pooling aggregates

the spatial information into a 96-dimensional feature vector, which is concatenated with a 16-dimensional device embedding learned from one-hot encoded device identifiers to distinguish different pipe diameters. The combined 112-dimensional representation is projected to the 32-dimensional latent representation z via a fully connected layer with dropout ($p = 0.2$). This architecture achieves $4 \times$ total spatial reduction while maintaining gradient flow through residual connections, enabling efficient multi-scale feature extraction from 3D flow field data.

Contrastive regression loss. As demonstrated in Fig. 4 above, the calibration factor value cannot be predicted or extrapolated between different configurations using simple rules or transformations. In order to learn the relevant features that predict the calibration factor of a newly installed flow meter, we train our model with a variety of operational configurations. To be able to distill only features that are essential and effective for the prediction of the calibration factor c , we deploy a loss function that is aimed at aligning the feature space according to the value of their label c . Inspired by image-based regression tasks in other domains [33], we train the representation learning encoder using a Ranked Contrastive Loss. This loss function extends the concept of contrastive learning from classification to regression tasks, ensuring that inputs with similar target values are represented similarly in the latent space. In this way, the representation space is ordered according to increasing values of the regression target variable c .

To calculate the contrastive regression loss, we contrast each anchor representation z_i with another representation z_j by defining the ranking set, including all representations whose calibration factor labels are further away from i than i is from j :

$$S_{ij} := \{z_k \mid k \neq i, d_{ik} \geq d_{ij}\}.$$

Here d_{ij} denotes the distance between the calibration factor labels of samples i and j , in our case the L_1 distance norm $d_{ij} = |c_i - c_j|$.

Given a batch of $2N$ input samples (in our case each sample includes 5×3 times 2D flow feature maps of a specific configuration), the representations $\{z_i\}_{i=1}^{2N}$ of the batch samples are extracted using the encoder network. The set of scalar calibration factor labels of the batch samples are denoted as $\{c_i\}_{i=1}^{2N}$. The ranked contrastive loss is then given by

$$\mathcal{L}_{\text{RNC}} = -\frac{1}{2N} \sum_{i=1}^{2N} \frac{1}{2N-1} \sum_{\substack{j=1 \\ j \neq i}}^{2N} \log P(z_j \mid z_i, S_{ij}).$$

Here $P(z_j \mid z_i, S_{ij})$ is the likelihood that the representation z_j is more similar to z_i than all elements in the ranking set S_{ij} ,

$$P(z_j \mid z_i, S_{ij}) = \frac{e^{\text{sim}(z_i, z_j)/\tau}}{\sum_{z_k \in S_{ij}} e^{\text{sim}(z_i, z_k)/\tau}},$$

where the similarity function $\text{sim}(z_i, z_j)$ quantifies the proximity of feature embeddings of the two representations. Here we use the cosine similarity $\text{sim}(z_i, z_j) = (z_i \cdot z_j) / (\|z_i\| \cdot \|z_j\|)$.

Minimizing the ranked contrastive loss aligns the similarity of embeddings with the similarity of their labels. As a result, the embedding space of sample representations is ordered in increasing order of the calibration factor values.

The regression module. Once training with the contrastive regression loss is completed, the embedding space z^* is ordered according to the target calibration factor values. This can be interpreted as following: during training the model learned to distill from the inputs only features that are informative for the prediction of the calibration factor. The next step is to train a regression model that maps the learned features onto the correct calibration factor value. The regression head processes the frozen latent representation through a four-layer fully connected network with exponentially decreasing dimensions ($32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 1$), where each hidden layer employs SELU activation and Gaussian dropout ($p = 0.2$) for regularization. Training employs Huber loss ($\delta = 0.001$) for robust regression, providing resilience against outliers while maintaining sensitivity to small prediction errors.

4.3.4. Train-test split

The original dataset (before augmentation) includes data collected from 6 different experimental setups, comprised of 3 different inlet disturbances each with 2 pipe diameters. Each experimental setup includes 70 measured calibration factor values in different device orientations, Reynolds numbers and inlet distances.

To test the generalization ability and applicability of the calibration factor prediction algorithm for new flow meter installations, we split the data such that the entire data from one experimental setup is not included in the training set and used only for testing. All experimental data from the remaining 5 setups is used for training and validation, with 5% of the samples randomly selected for validation.

5. Results

5.1. Calibration factor prediction performance

To evaluate the calibration factor prediction performance we compare the calibration factor value $\hat{c}_i$ predicted by the model for sample i , with the experimental measurement of the calibration factor c_i . We define the relative prediction residuals for each measured calibration factor as $\epsilon_i = (\hat{c}_i - c_i) / c_i$.

To evaluate the performance for a newly installed flow meter, we repeat the train-test split several times. Each time we test the regression performance on data from one out of the six experimental setups, which is then left out of the training data.

The results of the proposed model are contrasted with the results of a naive baseline, in which the calibration factor prediction for a certain inlet distance is estimated as the mean over all training values with the same inlet distance.

Fig. 8 shows the relative calibration factor prediction residuals (in %) for two different “leave-one-out” scenarios. Panel (a) shows the results when the data of the “Single bend DN50” setup is used for testing, and Panel (b) when the data of the “Double bend 3D DN50” setup is used for testing. In each panel the residuals are averaged over all the configurations with the same inlet distance. The resulting mean and confidence bands are shown as function of the inlet distance (on a log scale). The residuals of the test set (solid red) are compared with those of the training set (dashed blue) as well as with the naive baseline (dotted gray). Two horizontal black lines at $\pm 0.75\%$ indicate the measurement device’s target accuracy, serving as visual reference to assess whether residuals fall within the acceptable tolerance limits.

The results show that the prediction errors of the calibration factor fall well within the required specifications of 0.75% for all training conditions. These include a variety of inlet disturbances, device orientations, Reynolds numbers, and inlet distances. When testing the prediction for unseen setups the model achieves the required accuracy in the great majority of the measured conditions. As expected, for both of the testing setups, at very short inlet distances (close to the pipe bends) the strong non-linearity of the flow disturbance leads to higher prediction errors. However, despite the difficult transfer conditions (predicting for an unseen setup), these errors never exceed 1%. These remarkable results should be contrasted with our quotes of the effective calibration errors of up to 4.5%, that are typically reached under non-ideal flow conditions as the ones considered here. These results demonstrate the ability of the hybrid deep-learning approach to learn the essential properties of the flow that can serve as useful predictors of the calibration factor value, even though data from the specific setup has not been seen by the model during training. This is demonstrated here for two different unseen setups, which differ not only in their geometric properties (single inlet bending vs. three dimensional double bending) but also considerably differ in the complexity of the flow fields and their development along the pipe.

The comparison with a naive baseline confirms our findings regarding the strong generalization capability of the model: a simple

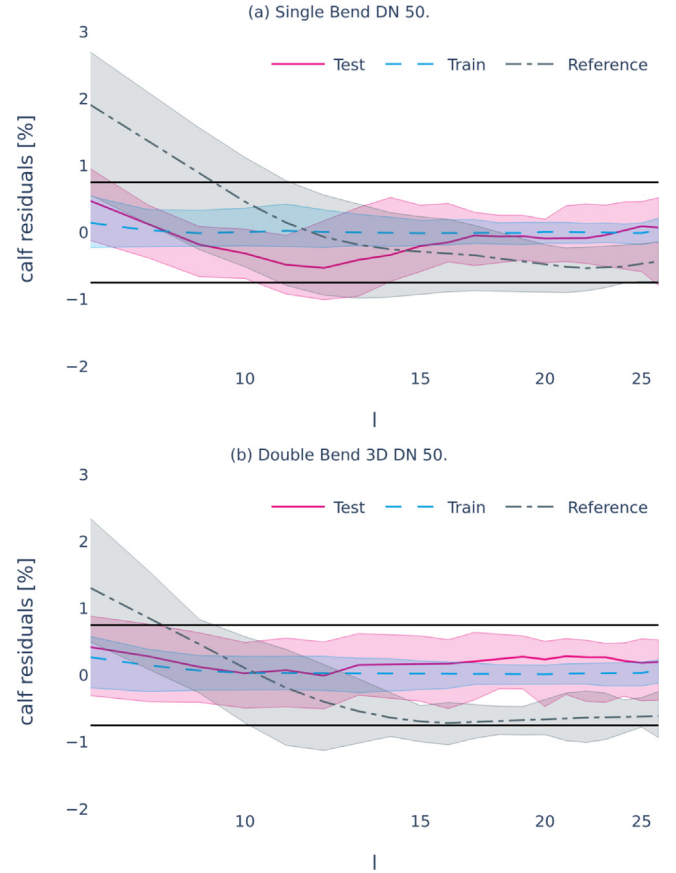


Fig. 8. Prediction performance results. The relative residuals ϵ of the calibration factor prediction are plotted against the inlet distance l for two test sets: (a) single bend DN50 and (b) double bend 3D DN50. Each plot shows the mean and 95% confidence band over all test residuals with the same inlet distance. The results of the proposed model (red) are contrasted with a naive baseline (gray). The training residuals are shown in blue for comparison.

averaging of the existing calibration factor measurements is not sufficient to accurately predict the calibration factor value for a new unseen setup. The naive baseline produced systematically biased calibration factor predictions, as well as very strong uncertainties, particularly at small to medium inlet distances, where different configurations tend to strongly differ from each other in their flow properties. Both the bias and the uncertainty are successfully mitigated by a model that captures the correct mapping between fundamental flow properties and the resulting calibration factor value. This is despite the fact that this relationship is highly non-trivial and not entirely understood.

5.2. Representation learning analysis

The proposed approach leverages the ability of deep learning to extract meaningful features from complex multi-dimensional data, in order to use these features for the calibration factor regression task. A crucial component of the model is the contrastive regression loss function, enforcing a target-ordered feature space.

The use of contrastive learning for classification tasks is well established [30]. Here we adopt an approach that generalizes contrastive learning for regression tasks in which the target variable is not categorical but continuous. The approach was introduced for regression tasks such as gaze direction as well as age prediction from face images [32, 33]. Here we adapt the approach for calibration factor regression from multi-channel 2D CFD flow fields. The contrastive regression loss is aimed at ordering the latent representation (embedding) space

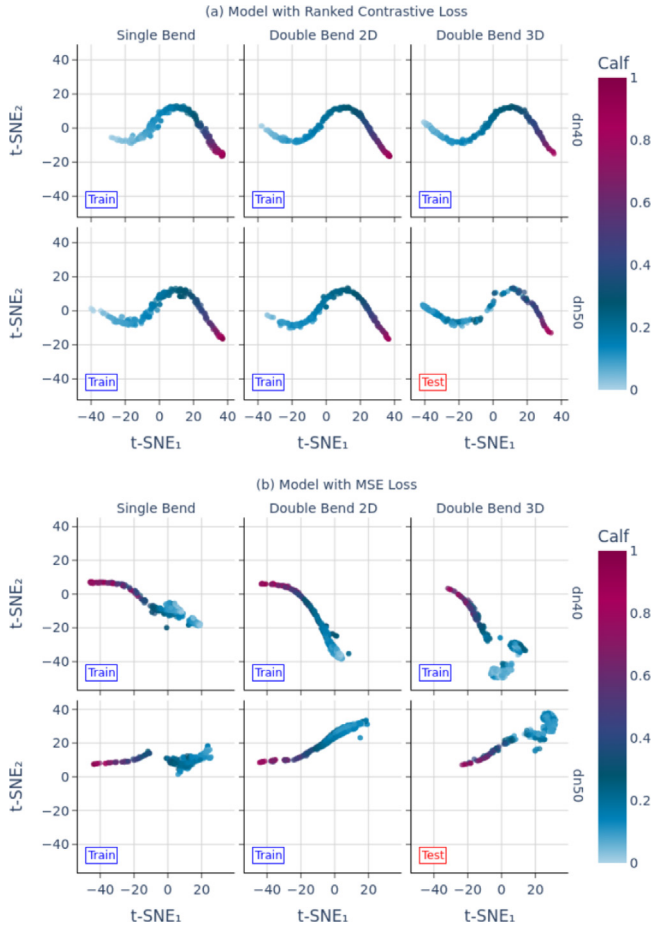


Fig. 9. Latent space visualization. t-SNE plots of the learned representation space of the model when trained with (a) Ranked Contrastive Loss and (b) MSE Loss. Each panel corresponds to one of the 6 setups, with the last setup (lower right panel) used for testing. Data points are colored by their calibration factor labels.

extracted from the flow maps according to the target variable, which is in this case the calibration factor value.

To demonstrate the effect of the contrastive loss on the learned embedding space we visualize the embedding space with and without the contrastive loss. For the visualization of the embedding space we use the standard t-SNE dimension reduction technique [49] and plot each sample as a point in the reduced 2D-space of the first and second t-SNE dimensions.

Fig. 9 shows t-SNE plots of the latent space of the trained model with (a) and without (b) contrastive loss, using a standard MSE loss instead. The color of each point corresponds to its true calibration factor (calf) value. In this example, the testing setup was the double bend 3D DN50, with the other 5 setups used for training. The six setups are shown in separate panels to allow a direct comparison of the latent space between different setups. It is evident that the effect of the contrastive loss on the embedding space is crucial: only with contrastive learning, the embeddings are ordered according to their calibration factor values, where samples of similar calibration factor are represented close to each other in representation space. Moreover, only the contrastive loss enables the same latent space ordering for all 6 setups, indicating that the model learns features that have predictive power of the calibration factor, ignoring features that discriminate the different setups from each other. In other words, the contrastive regression loss allows the model to learn latent features that are relevant for the calibration factor prediction and to ignore features that are irrelevant for this task.

It is important to note that the embedding space is correctly ordered not only for the training data, but also for the samples of the test setup (bottom right), that have not been seen during training. This implies that the model learns to generalize the feature extraction out of the training domain, despite the fact that none of the samples of the test setup was used for training. As a consequence, the regression task of mapping the latent space to the calibration factor value now becomes a simple task, obtained by a small fully connected network. The successful calibration factor prediction for a new unseen setup is the key achievement of the proposed method, as it allows to train on a small amount of experimental data from known setups and accurately predict the calibration factor for newly installed setups.

5.3. Interpretability and feature importance

The complex nature of the input and the black-box nature of the modeling approach encourage us to apply post-hoc approaches for deep learning model interpretability. In particular we aim at understanding what the model focusses on when it learns useful latent representations for the calibration factor prediction task. This can enhance the understanding of the physical variables that dominate the calibration factor dependence under complex conditions, at which purely physics-based calibration factor prediction is highly inaccurate.

Analyzing which parts of the input are relevant for the model output is commonly done using saliency maps and was first suggested for classification tasks [50]. The main idea of saliency maps is to visualize the gradients of the model output with respect to the single input pixels, and in this way to highlight the regions of the input, where small changes would cause the largest change in the output. In our case the input is composed of $5 \times 3 \times 2D$ maps, such that the calculated saliency maps end up having the same shape.

In order to enhance interpretability and reduce the noise of the saliency maps we calculated the saliency gradients, followed by a sequence of several well established smoothing techniques. In particular we followed the Guided Integrated Gradients method of [51] with a zero baseline (justified for inputs that indicate differences from the undisturbed flow), followed by path interpolation [52], guided back-propagation [53], and Gaussian smoothing [54]. To focus on the importance attribution for the latent representation learning, the saliency maps in our implementation correspond to gradients of the L_2 norm of the latent representation $\|z\|_2$ instead of the standard gradients of the regression output variable itself.

Fig. 10 displays the input maps and the saliency maps for two selected configurations. For each configuration, the upper row shows the original 5 input channels (for simplicity we focus on the I_j features) and the lower row shows the calculated saliency maps, where 1 indicates the highest importance. For each saliency map we calculated its relative importance (in %) by dividing its total importance score (summed over all pixels) by the total score of all channels together, and the result is indicated at the bottom of each map.

A close observation of the saliency maps in Fig. 10 reveals that 3 of the channels are assigned a high relative importance: the downstream velocity, mean angles, and directional variability. This result is noticeable for both of the configurations shown in the figure (single and double bend 2D), and is consistent with physical intuition.

A second observation from the saliency maps is that the bright (high attention) areas are concentrated mostly near the pipe walls (represented at the right edge of the maps). This is consistent with fluid dynamic expectations, as these near-wall regions, characterized by high shear stress and boundary layer effects, are the primary drivers influencing the calibration factor correction. The fact that the model independently identifies these physically critical regions as highly relevant validates its ability to extract meaningful physical features from the flow field, rather than relying on arbitrary correlations. While these results provide a strong qualitative foundation, future studies should focus on gaining a more quantitative understanding of the specific flow properties that determine the calibration factor.

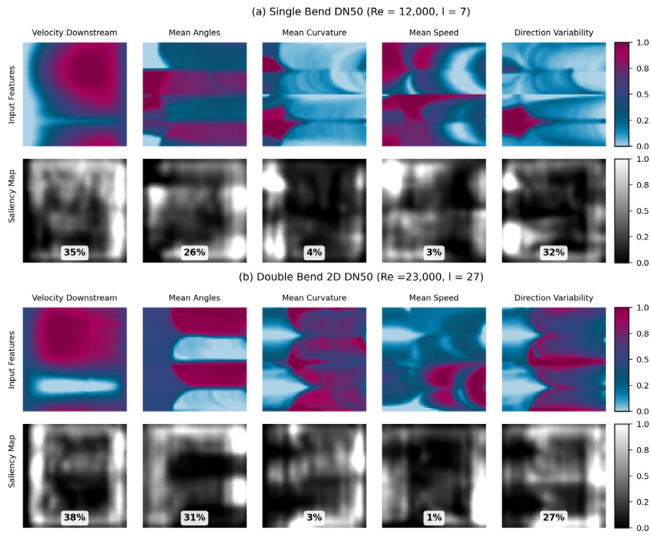


Fig. 10. Model interpretation. 5 input channels (color) and their corresponding saliency maps (black & white) are shown for two testing configurations (a) Single Bend DN50 (Re = 12,000, $l = 7$), and (b) Double Bend 2D DN50 (Re = 23,000, $l = 27$). The relative importance of each channel is indicated at the bottom of the saliency map.

5.4. Ablation study

The proposed modeling approach contains several ingredients which are essential to guarantee the high performance of the flow measurement error prediction. A central ingredient, discussed at length in Section 5.2 is the contrastive regression loss function. As shown, reverting to a simple MSE loss fails to capture the informative features of the flow fields that allow for accurate calf prediction. Below we identify two additional ingredients of our approach, and demonstrate their contribution to the prediction performance.

Relative flow fields as inputs. The proposed model is trained with 2D flow field maps that are obtained by subtracting a reference flow field from the measured one. The reference flow field is the steady flow in a straight pipe under the same Reynolds number and at the same inlet distance. To demonstrate the effect of the relative compared to the absolute flow field, we train a model with the absolute flow field maps as inputs. The prediction results of the two model variants are shown in Fig. 11(a) for the “double bend 3D” setup test data (used in both cases only for testing). It is observed that the results based on absolute fields (yellow) fluctuate more strongly in their prediction accuracy, with prediction residuals that are occasionally above the 0.75% target. Thus, training with the relative flow field maps (red) significantly increases the calf prediction accuracy.

Flow dynamics as inputs. As explained in Section 4.3.2, the inputs to the model at inlet distance l_j comprise three flow field maps for each of the 5 channels, corresponding to transverse cuts of the pipe at the three consecutive inlet distances l_{j-2}, l_{j-1}, l_j . This feature engineering is motivated by the physical understanding that the calf prediction error is strongly influenced by dynamic properties of the flow, that are reflected in flow modifications along the pipe axis. To test this hypothesis we performed an ablation study, training the model with a flow field at inlet distance l_j without the inputs of additional distances. The results of this comparison are shown in Fig. 11(b) for the “double bend 3D” setup test data (used only for testing). We observe that the dynamic information is useful for accurate calf predictions particularly in intermediate inlet distances, where discarding it (yellow) leads to increased prediction residuals compared to the proposed dynamic inputs (red).

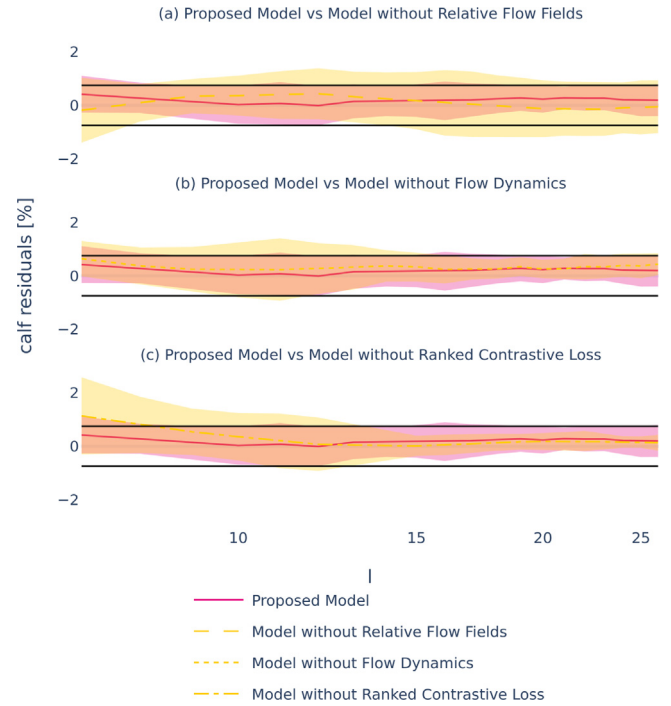


Fig. 11. Ablation study. The relative calf prediction residuals for a test setup (double bend 3D DN50) are plotted against the inlet distance l , comparing the proposed model (red) with a competing model version (yellow). Three model versions are shown: (a) using absolute instead of relative flow fields as inputs (b) using no flow dynamics as input (c) training with a standard MSE loss instead of ranked contrastive loss.

Ranked contrastive loss. As elaborated in Section 5.2, an important model ingredient is the ranked contrastive loss, which enables the alignment of the latent representations according to the target calf value. Replacing the contrastive loss with a standard MSE loss, thus directly minimizing the prediction error without a prior latent space alignment, is a third ablation study we performed. As shown in Fig. 9(b) in the absence of the contrastive loss, the latent representations of the test data depend strongly on the setup (inlet disturbance and diameter). As a result, predicting the calf for a new installation becomes challenging and inaccurate. The prediction errors with (red) and without (yellow) contrastive loss are shown in Fig. 11(c) for the “double bend 3D” test data. The prediction residuals without contrastive loss are significantly higher at low inlet distances (close to the disturbance), where the disturbance type is most important. This demonstrates that the latent space alignment achieved by the contrastive loss is essential to obtain accurate calf predictions for newly installed sensor configurations.

5.5. Computational setup and hyperparameter optimization

Hardware and software infrastructure. All experiments were conducted on a server equipped with an NVIDIA H100 GPU (80 GB HBM3). The implementation utilized PyTorch 2.5.1 [55] and PyTorch Lightning 2.4.0 [56] with automatic mixed precision training to optimize memory usage and computational efficiency.

Hyper-parameter optimization. To maximize model performance, systematic hyper-parameter tuning was conducted using the open-source framework Optuna 4.4.0 [57]. Specifically, the Tree-structured Parzen Estimator (TPE) was employed as the optimization strategy, enabling guided exploration of the hyper-parameter space while prioritizing promising configurations. The optimal hyper-parameter set was selected based on validation performance, with multiple independent

Table 1
Hyper-parameter search (optimal values marked with *).

Hyper-parameter	Type	Search range/values
Feature Extractor		
Dropout Convolution Layer	Float	[0.1, 0.2*, 0.3]
Dropout FC-Layer	Float	[0.1, 0.2*, 0.3]
Encoder Channels	Tuple	[(8, 16, 32), (16, 32, 64), (32, 64, 96)*]
Latent Space Dimension	Integer	[16, 32*, 64]
Regression Model		
Depth	Integer	[3, 4*, 5]
Dropout	Float	[0.1, 0.2*, 0.3]
Delta	Float	[0.1, 0.005, 0.001*]

Table 2
ML training runtime benchmarks.

Metric	Value
Total number of epochs	50
Time per epoch	45 s
Total training time	37.5 min

trials conducted to ensure statistical robustness and reproducibility of results. Table 1 shows the hyper-parameter search ranges and optimal values (marked with *).

5.6. Computational performance and runtime analysis

To demonstrate the operational viability of the proposed approach, we provide detailed runtime benchmarks for both the machine learning training phase and the inference stage.

Training performance. The neural network was trained for 50 epochs on NVIDIA H100. Table 2 summarizes the training runtime metrics.

Inference performance. A critical requirement for operational deployment is real-time prediction capability. Our trained model achieves inference times of *less than 5 ms per prediction*, enabling seamless integration into time-critical applications and real-time monitoring systems.

CFD performance. The CFD simulations for generating the training dataset were performed using StarCCM+ 2310 in double precision mode on a HPC cluster using AMD EPYC 7742 nodes with 64 cores per node. The CFD process including meshing, computation and postprocessing was fully automated and required minimal user input. Due to the different configurations and Reynolds number-dependent meshing simulation times varied widely for different configurations, but each simulation required less than one hour on average with the meshing process being completed in less than one minute. All simulations were performed using 128 CPU cores.

As the available HPC infrastructure was not a limiting factor for these calculations, no additional runtime tests were performed. We estimate that the computation time per simulation could be further reduced by at least 50%.

These benchmarks confirm that the proposed method achieves operationally viable computation times suitable for practical deployment.

6. Conclusion

Systematic measurement errors pose a long-standing challenge to high precision flow meters. For flow meters based on dynamic measurement principles, such as vortex-based meters, the problem is aggravated by the impracticability of error correction based on CFD simulations alone. On the other hand, purely data-driven error correction may be suitable for existing measurement setups, but cannot provide accurate enough results for operational applications with previously unseen sensor installations.

In this paper we presented a novel hybrid approach to flow measurement error correction based on a combination of CFD simulation outputs and a deep representation learning model. The method exploits the CFD tool in steady, computationally tractable regimes, and complements the missing dynamics by learning CFD features that have predictive power of measurement errors. After being trained on lab data, the model is capable of accurately predicting the systematic measurement errors for newly installed setups, unseen during training, despite their high physical complexity. This is enabled due to the use of ranked contrastive learning, which aligns the latent CFD-driven feature space with the value of the measurement calibration factor. The use of contrastive regression for technical images (and in particular for CFD flow fields) is demonstrated here for the first time.

The model shows a particularly strong performance in challenging flow regimes, including low Reynolds number conditions and regions close to the pipe inlet, where alternative methods often struggle with accuracy and computational barriers. Furthermore, the machine learning framework provides valuable insights into the physical influences governing flow behavior across different setup geometries and flow parameters, contributing to targeted sensor design improvements and optimization strategies.

The proposed framework illustrates how advanced deep learning models can both substitute for and augment traditional simulation tools such as CFD, particularly in highly complex and strongly non-linear regimes where conventional simulations encounter fundamental limitations in accuracy and computational efficiency. We therefore believe that the potential applications of this framework extend far beyond mere error correction in flow measurement devices, opening promising avenues for broader use.

Following these promising results, our future research will focus on expanding the data set and validating the approach under a broader diversity of measurement setups and operational conditions. We aim to investigate material property variations, temperature effects beyond Reynolds number encoding, and multiphase flow regimes, which would require enhanced CFD modeling and additional experimental validation data. Furthermore, deeper analysis of the automated feature selection mechanism could provide additional insights into the physical drivers of measurement errors and guide targeted sensor design improvements.

CRedit authorship contribution statement

Stephan Wernli: Writing – original draft, Visualization, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Marc Hollmach:** Writing – review & editing, Visualization, Methodology, Investigation, Formal analysis, Data curation. **Christian Franzmann:** Writing – review & editing, Visualization, Software, Methodology, Formal analysis, Data curation. **Daniel Kessler:** Writing – review & editing, Software, Data curation, Conceptualization. **Henrique Fernandes:** Writing – review & editing, Supervision. **Lilach Goren Huber:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used gpt-oss in order to refine and improve the textual content of this paper. gpt-oss was utilized to enhance the clarity, coherence, and linguistic quality of the written material. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published article.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Stephan Wernli, Marc Hollmach, Christian Franzmann and Daniel Kessler report financial support, equipment, drugs, or supplies were provided by Endress+Hauser Flowtec AG. These authors also report a relationship with Endress+Hauser Flowtec AG that includes: employment. Lilach Goren Huber reports financial support was provided by ZHAW University of Applied Sciences. Stephan Wernli, Marc Hollmach, Christian Franzmann, Daniel Kessler and Lilach Goren Huber have patent pending #DE 102025116816.4 assigned to Endress+Hauser Flowtec AG and Zürcher Hochschule für Angewandte Wissenschaften (ZHAW). If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to express their gratitude to Endress+Hauser Flow and Endress+Hauser InfoServe for supporting the study. The authors would also thank Dr. Rebecca Page for valuable discussions and inputs.

Appendix. CFD methodology

The resultant flow profiles of three flow disturbance configurations, corresponding to the experimental configurations in Section 4.1, were simulated using StarCCM+. The objective of these simulations is to obtain a flow field that represents the inflow condition for the vortex flow meter. The flow field is then used as input for the deep representation learning model.

Each disturbance configuration is set up with a straight pipe inlet section with a length of 10 pipe diameters and a straight pipe outlet section with a length of 100D. Flow disturbance profiles are extracted downstream of the disturbance in 1D increments. As only steady incompressible flows are considered within the scope of this work, the Reynolds number $Re = \rho U D / \mu$ is the only independent variable besides the geometric position [58]. Therefore the variables ρ , U and D are all set to the numeric value of 1 (using the appropriate units in the SI unit system) in order to obtain non-dimensional flow variables directly from the simulation results. The desired Reynolds number is set by adjusting the dynamic viscosity μ .

For each configuration, profiles corresponding to the Reynolds numbers of the experiments are simulated. Using the StarCCM+ meshing tool, a parametrized computational mesh consisting of a prism layer and a polyeder volume mesh is generated. The initial wall layer thickness is $\frac{\sqrt{8D}}{\sqrt{\lambda Re}}$ (with $\lambda = 0.02$, assuming hydraulically smooth pipes), corresponding to a theoretical y^+ value of 1. Using a growth ratio of 1.2, the prism layer is built until the target volume mesh element size is reached. This size is set to 3% of the pipe diameter, following a full mesh independence study. The mesh element size is refined by a factor of 2 in the immediate vicinity of the disturbance.

Steady-state RANS (*Reynolds-averaged Navier–Stokes*) calculations are performed using StarCCM+'s segregated solver and the MUSCL 3rd order upwind discretization scheme, limiting spurious oscillations in the solution. As turbulence model a Reynolds Stress Transport model is chosen, namely the Linear Pressure-Strain Two-Layer model [59]. The velocity profile for an infinitely long pipe at the respective Reynolds number is applied as inlet condition. This profile is calculated using the same simulation setup, but a different (much smaller) mesh, representing only a small slice of a pipe. The outlet boundary condition does not allow backflow.

Data availability

The authors do not have permission to share data.

References

- [1] G.S. de Aquino, R.S. Martins, M.F. Martins, R.F. Ramos, An overview of computational fluid dynamics as a tool to support ultrasonic flow measurements, *Metrology* 5 (1) (2025) 1–26, <http://dx.doi.org/10.3390/metrology5010011>.
- [2] M. Takamoto, H. Utsumi, N. Watanabe, Y. Terao, Installation effects on vortex shedding flowmeters, *Flow Meas. Instrum.* 4 (4) (1993) 277–285, [http://dx.doi.org/10.1016/0955-5986\(93\)90036-1](http://dx.doi.org/10.1016/0955-5986(93)90036-1), URL <https://www.sciencedirect.com/science/article/pii/0955598693900361>.
- [3] J. Miao, C. Wu, C. Hu, J. Chou, A study on signal quality of a vortex flowmeter downstream of two elbows out-of-plane, *Flow Meas. Instrum.* 13 (3) (2002) 75–85, [http://dx.doi.org/10.1016/S0955-5986\(02\)00027-4](http://dx.doi.org/10.1016/S0955-5986(02)00027-4), URL <https://www.sciencedirect.com/science/article/pii/S0955598602000274>.
- [4] M. Straka, A. Weißengrünner, C. Koglin, C. Höhne, S. Schmelter, Simulation uncertainty for a virtual ultrasonic flow meter, *Metrology* 2 (3) (2022) 335–359, <http://dx.doi.org/10.3390/metrology2030021>.
- [5] P. Papatthanasou, A Hybrid Numerical Method for 3D Multiphysics Modeling of Ultrasonic Transit-Time Flowmeters: Including Sound Propagation in Real Pipe Flows (Ph.D. thesis), The University of Bergen, 2023, URL <https://hdl.handle.net/11250/3095177>. Simulations of Piezoelectricity, Acoustics, Coupled with CFD (SimPAC2).
- [6] L. Chen, C. Xia, Z. Zhao, H. Fu, Y. Chen, AI-driven sensing technology: Review, *Sensors* 24 (10) (2024) <http://dx.doi.org/10.3390/s24102958>, URL <https://www.mdpi.com/1424-8220/24/10/2958>.
- [7] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* 52 (Volume 52, 2020) (2020) 477–508, <http://dx.doi.org/10.1146/annurev-fluid-010719-060214>, URL <https://www.annualreviews.org/content/journals/10.1146/annurev-fluid-010719-060214>.
- [8] C. Liu, C. Zhao, Y. Wang, H. Wang, Machine-learning-based calibration of temperature sensors, *Sensors* 23 (17) (2023) <http://dx.doi.org/10.3390/s23177347>, URL <https://www.mdpi.com/1424-8220/23/17/7347>.
- [9] S. Koziel, A. Pietrenko-Dabrowska, M. Wojcikowski, B. Pankiewicz, On memory-based precise calibration of cost-efficient NO2 sensor using artificial intelligence and global response correction, *Knowl.-Based Syst.* 290 (2024) 111564, <http://dx.doi.org/10.1016/j.knsys.2024.111564>, URL <https://www.sciencedirect.com/science/article/pii/S0950705124001990>.
- [10] C. Bachechi, F. Rollo, L. Po, HypeAIR: A novel framework for real-time low-cost sensor calibration for air quality monitoring in smart cities, *Ecol. Inform.* 81 (2024) 102568, <http://dx.doi.org/10.1016/j.ecoinf.2024.102568>, URL <https://www.sciencedirect.com/science/article/pii/S1574954124001109>.
- [11] I. Podder, T. Fischl, U. Bub, Smart calibration and monitoring: leveraging artificial intelligence to improve MEMS-based inertial sensor calibration, *Complex Intell. Syst.* 10 (6) (2024) 7451–7474, <http://dx.doi.org/10.1007/s40747-024-01531-y>.
- [12] X. Zhao, P. Li, K. Xiao, X. Meng, L. Han, C. Yu, Sensor drift compensation based on the improved LSTM and SVM multi-class ensemble learning models, *Sensors* 19 (18) (2019) <http://dx.doi.org/10.3390/s19183844>, URL <https://www.mdpi.com/1424-8220/19/18/3844>.
- [13] D. Badawi, A. Agambayev, S. Ozev, A. Enis Çetin, Discrete cosine transform based causal convolutional neural network for drift compensation in chemical sensors, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2021, pp. 8012–8016, <http://dx.doi.org/10.1109/ICASSP39728.2021.9414512>.
- [14] C. Mai, Z. Wang, L. Chen, Y. Huang, M. Li, A. Shirazi, A. Altaee, J.L. Zhou, Field-based calibration and operation of low-cost sensors for particulate matter by linear and nonlinear methods, *Atmospheric Pollut. Res.* 16 (12) (2025) 102676, <http://dx.doi.org/10.1016/j.apr.2025.102676>, URL <https://www.sciencedirect.com/science/article/pii/S1309104225002788>.
- [15] A. Krayden, M. Avraham, H. Ashkar, T. Blank, S. Stolyarova, Y. Nemirovsky, Tinyml-based real-time drift compensation for gas sensors using spectral-temporal neural networks, *Chemosensors* 13 (7) (2025) <http://dx.doi.org/10.3390/chemosensors13070223>, URL <https://www.mdpi.com/2227-9040/13/7/223>.
- [16] Z. Yang, W. Wang, X. Liu, X. Chen, X. Li, X. Lu, Research on measurement error distribution and optimization measurement method of clamp-on ultrasonic flowmeter in downstream pipeline disturbance, *Sensors* 25 (13) (2025) <http://dx.doi.org/10.3390/s25134011>, URL <https://www.mdpi.com/1424-8220/25/13/4011>.
- [17] T. Savioli, M. Pampanini, G. Visani, L. Esposito, C.A. Rinaldini, Engine mass flow estimation through neural network modeling in semi-transient conditions: A new calibration approach, *Fluids* 9 (10) (2024) <http://dx.doi.org/10.3390/fluids9100239>, URL <https://www.mdpi.com/2311-5521/9/10/239>.

- [18] T. Murata, K. Fukami, K. Fukagata, Nonlinear mode decomposition with convolutional neural networks for fluid dynamics, *J. Fluid Mech.* 882 (2020) A13, <http://dx.doi.org/10.1017/jfm.2019.822>.
- [19] D. Kochkov, J.A. Smith, A. Alieva, Q. Wang, M.P. Brenner, S. Hoyer, Machine learning-accelerated computational fluid dynamics, *Proc. Natl. Acad. Sci.* 118 (21) (2021) e2101784118, <http://dx.doi.org/10.1073/pnas.2101784118>, arXiv: <https://www.pnas.org/doi/pdf/10.1073/pnas.2101784118>, URL <https://www.pnas.org/doi/abs/10.1073/pnas.2101784118>.
- [20] Z. Liu, R. Han, M. Zhang, Y. Zhang, H. Zhou, G. Wang, G. Chen, An enhanced hybrid deep neural network reduced-order model for transonic buffet flow prediction, *Aerosp. Sci. Technol.* 126 (2022) 107636, <http://dx.doi.org/10.1016/j.ast.2022.107636>, URL <https://www.sciencedirect.com/science/article/pii/S1270963822003108>.
- [21] A. Abucide-Armas, K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, A. Teso-Fz-Beño, Convolutional neural network predictions for unsteady Reynolds-averaged Navier-Stokes-based numerical simulations, *J. Mar. Sci. Eng.* 11 (2) (2023) <http://dx.doi.org/10.3390/jmse11020239>, URL <https://www.mdpi.com/2077-1312/11/2/239>.
- [22] N. Geneva, N. Zabarar, Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks, *J. Comput. Phys.* 403 (2020) 109056, <http://dx.doi.org/10.1016/j.jcp.2019.109056>, URL <https://www.sciencedirect.com/science/article/pii/S0021999119307612>.
- [23] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707, <http://dx.doi.org/10.1016/j.jcp.2018.10.045>, URL <https://www.sciencedirect.com/science/article/pii/S0021999118307125>.
- [24] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: A review, 2021, arXiv:2105.09506, URL <https://arxiv.org/abs/2105.09506>.
- [25] M. Rom, Physics-informed neural networks for the Reynolds equation with cavitation modeling, *Tribol. Int.* 179 (2023) 108141, <http://dx.doi.org/10.1016/j.triboint.2022.108141>, URL <https://www.sciencedirect.com/science/article/pii/S0301679X22007125>.
- [26] X. Jin, S. Cai, H. Li, G.E. Karniadakis, Nfnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 426 (2021) 109951, <http://dx.doi.org/10.1016/j.jcp.2020.109951>, URL <https://www.sciencedirect.com/science/article/pii/S0021999120307257>.
- [27] P.-H. Chiu, J.C. Wong, C. Ooi, M.H. Dao, Y.-S. Ong, CAN-PINN: A fast physics-informed neural network based on coupled-automatic-numerical differentiation method, *Comput. Methods Appl. Mech. Engrg.* 395 (2022) 114909, <http://dx.doi.org/10.1016/j.cma.2022.114909>, URL <https://www.sciencedirect.com/science/article/pii/S0045782522001906>.
- [28] C. Cheng, G.-T. Zhang, Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems, *Water* 13 (4) (2021) <http://dx.doi.org/10.3390/w13040423>, URL <https://www.mdpi.com/2073-4441/13/4/423>.
- [29] Z. Xiang, W. Peng, W. Yao, X. Liu, X. Zhang, Physics-informed neural implicit flow neural network for parametric PDEs, *Neural Netw.* 185 (2025) 107166, <http://dx.doi.org/10.1016/j.neunet.2025.107166>, URL <https://www.sciencedirect.com/science/article/pii/S0893608025000450>.
- [30] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: *International Conference on Machine Learning*, PmlR, 2020, pp. 1597–1607.
- [31] H. Hu, X. Wang, Y. Zhang, Q. Chen, Q. Guan, A comprehensive survey on contrastive learning, *Neurocomputing* 610 (2024) 128645.
- [32] Y. Wang, Y. Jiang, J. Li, B. Ni, W. Dai, C. Li, H. Xiong, T. Li, Contrastive regression for domain adaptation on gaze estimation, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022*, pp. 19376–19385.
- [33] K. Zha, P. Cao, J. Son, Y. Yang, D. Katabi, Rank-n-contrast: Learning continuous representations for regression, *Adv. Neural Inf. Process. Syst.* 36 (2023) 17882–17903.
- [34] X. Ke, H. Xu, X. Lin, W. Guo, Two-path target-aware contrastive regression for action quality assessment, *Inform. Sci.* 664 (2024) 120347.
- [35] X. Yu, Y. Rao, W. Zhao, J. Lu, J. Zhou, Group-aware contrastive regression for action quality assessment, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021*, pp. 7919–7928.
- [36] A. Ragano, J. Skoglund, A. Hines, SCOREQ: Speech quality assessment with contrastive regression, *Adv. Neural Inf. Process. Syst.* 37 (2024) 105702–105729.
- [37] S. Ge, H. Gu, W. Su, A. Lönnqvist, O. Antropov, A novel semisupervised contrastive regression framework for forest inventory mapping with multisensor satellite data, *IEEE Geosci. Remote. Sens. Lett.* 20 (2023) 1–5.
- [38] Q. Zhu, Z. Zhou, Y. Li, R. Yan, Contrastive BiLSTM-enabled health representation learning for remaining useful life prediction, *Reliab. Eng. Syst. Saf.* 249 (2024) 110210.
- [39] T.v. Kármán, Ueber den Mechanismus des Widerstandes, den ein bewegter Körper in einer Flüssigkeit erfährt, *Nachrichten Von Ges. Wiss. Zu Gött. Mathematisch-Physikalische Kl.* 1911 (1911) 509–517, URL <http://eudml.org/doc/58812>.
- [40] Endress+Hauser, Das Wirbelzähler-Durchflussmessprinzip, 2024, URL <https://www.ch.endress.com/de/support-overview/learnplattform/messprinzip-durchflussmessung-vortex>, Lernplattform für Messtechnik.
- [41] Siemens Digital Industries Software, Simcenter STAR-CCM+, version 2310, Siemens 2023.
- [42] G.M. Nielson, R. Ramaraj, Interpolation over a sphere based upon a minimum norm network, *Comput. Aided Geom. Design* 4 (1) (1987) 41–57, [http://dx.doi.org/10.1016/0167-8396\(87\)90023-9](http://dx.doi.org/10.1016/0167-8396(87)90023-9), URL <https://www.sciencedirect.com/science/article/pii/0167839687900239>, Topics in CAGD.
- [43] M.K. Rashid, K.N. Abed, M. Alharbi, M.W. Muayad, T. Alkharusi, A. Ameen, Integrating machine learning and CFD for enhanced trailing edge serration design on a NACA 0012 wind turbine blade, *Int. J. Thermofluids* 30 (2025) 101446, <http://dx.doi.org/10.1016/j.ijft.2025.101446>, URL <https://www.sciencedirect.com/science/article/pii/S2666202725003921>.
- [44] Y. Li, J. Chang, C. Kong, Z. Wang, Flow field reconstruction and prediction of the supersonic cascade channel based on a symmetry neural network under complex and variable conditions, *AIP Adv.* 10 (6) (2020) 065116, <http://dx.doi.org/10.1063/5.0008889>.
- [45] M.Z. Yousif, L. Yu, S. Hoyas, R. Vinuesa, H. Lim, A deep-learning approach for reconstructing 3D turbulent flows from 2D observation data, *Sci. Rep.* 13 (1) (2023) 2529, <http://dx.doi.org/10.1038/s41598-023-29525-9>.
- [46] O. Sallam, M. Fürth, Dual scale residual-network for turbulent flow sub grid scale resolving: A prior analysis, *Comput. & Fluids* 292 (2025) 106592, <http://dx.doi.org/10.1016/j.compfluid.2025.106592>, URL <https://www.sciencedirect.com/science/article/pii/S0045793025000520>.
- [47] K. Hara, H. Kataoka, Y. Satoh, Learning spatio-temporal features with 3d residual networks for action recognition, in: *Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017*, pp. 3154–3160.
- [48] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016*, pp. 770–778.
- [49] L. van der Maaten, G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.* 9 (2008) 2579–2605.
- [50] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, 2013, arXiv preprint [arXiv:1312.6034](https://arxiv.org/abs/1312.6034).
- [51] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: *International Conference on Machine Learning, PMLR, 2017*, pp. 3319–3328.
- [52] A. Jha, J.K. Aicher, M.R. Gazzara, D. Singh, Y. Barash, Enhanced integrated gradients: improving interpretability of deep learning models using splicing codes as a case study, *Genome Biol.* 21 (1) (2020) 1–22.
- [53] A. Kaphishnikov, T. Bolukbasi, F. Viégas, M. Terry, XRAI: Better attributions through regions, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021*, pp. 4948–4957.
- [54] D. Smilkov, N. Thorat, B. Kim, F. Viégas, M. Wattenberg, SmoothGrad: removing noise by adding noise, 2017, arXiv preprint [arXiv:1706.03825](https://arxiv.org/abs/1706.03825).
- [55] J. Ansel, E. Yang, H. He, N. Gimpelstein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M.Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, S. Chintala, PyTorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation, *ASPLOS'24*, in: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, vol. 2, ACM, 2024, <http://dx.doi.org/10.1145/3620665.3640366>, URL <https://docs.pytorch.org/assets/pytorch2-2.pdf>.
- [56] W. Falcon, The PyTorch Lightning team, PyTorch Lightning, 2019, <http://dx.doi.org/10.5281/zenodo.3828935>, URL <https://github.com/Lightning-AI/lightning>.
- [57] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019*.
- [58] H. Oertel, M. Böhle, T. Reviol, *Strömungsmechanik*, Seventh ed. SpringerVieweg, Wiesbaden, 2015.
- [59] C. Speziale, S. Sarkar, T. Gatski, Modelling the pressure-strain correlation of turbulence: an invariant dynamical systems approach, *J. Fluid Mech.* 227 (1991) 245–272.



DSPACE

<https://dspace.org/>

Enhancing vortex-flow-meter precision using physics-informed contrastive learning

Wernli, Stephan; Hollmach, Marc; Franzmann, Christian; Kessler, Daniel; Fernandes, Henrique ...

2026-06

Elsevier

<https://doi.org/10.1016/j.flowmeasinst.2026.103207>

Wernli S, Hollmach M, Franzmann C, et al., (2026) Enhancing vortex-flow-meter precision using physics-informed contrastive learning. Flow Measurement and Instrumentation, Volume 109, June 2026, Article number 103207