Journal of Process Control 165 (2026) 103785

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

# A unified utility function-based framework for prior-informed surrogate modeling

András Edvy [a], Alex Kummer [a,b,*], János Abonyi [a,b]

[a] Department of Process Engineering, University of Pannonia, Egyetem st. 10, H-8200 Veszprem, Hungary
[b] HUN-REN-PE Complex Systems Monitoring Research Group, University of Pannonia, Egyetem st. 10, H-8200 Veszprem, Hungary

ARTICLE INFO

*Keywords:*
Utility-function framework
Hybrid modeling
Physics-informed learning
Surrogate models
Thermal energy systems
Data-efficient AI

ABSTRACT

Modern energy infrastructures operate under nonlinear, time-varying conditions and therefore require reliable surrogate models for optimization, predictive control and health monitoring. Existing approaches are brittle: first-principles simulators degrade under parameter drift and imperfect multiscale physics, while purely data-driven methods require large datasets and extrapolate poorly. We propose the Utility-balanced Prior-informed Model (UPriMo), a general training objective that integrates prior knowledge — such as physical laws, topology constraints, operating envelopes and expert heuristics — directly into the loss of any differentiable learner. UPriMo maps each prior residual to a bounded utility and minimizes data misfit plus utility shortfall. This yields automatic trade-offs between data and priors without manual weighting and recovers physics-informed learning as special cases. We demonstrate UPriMo on two energy-relevant problems. First, for stationary laminar pipe flow, a neural surrogate trained on five noisy velocity measurements satisfies Navier–Stokes, boundary and monotonicity constraints, and reduces mean-squared error by an order of magnitude compared with data-driven baselines. Second, for electric water-heater dynamics, nonlinear autoregressive models with exogenous inputs (NARX) augmented with steady-state and monotonicity priors remain stable over 1400-step free runs, whereas unconstrained models diverge. Across both problems, UPriMo improves accuracy, noise robustness and physical plausibility while remaining compatible with standard ML frameworks. Because the utilities are problem-independent, the framework supports a single recipe from component-level surrogates to system-level AI-based forecasting and control in future energy systems.

## 1. Introduction

Thermal power plants, district-heating networks, industrial furnaces and heat-recovery units form the backbone of the process industry and account for a major share of global final-energy use [1,2]. Operating these assets efficiently requires dynamic models that are accurate enough for load-flexible operation, emission reduction and advanced process optimization, yet simple enough to be deployed in estimation, forecasting and control. Such systems exhibit nonlinear thermodynamics, distributed-parameter transport phenomena and time-varying operating conditions. Parameter drift caused by fouling in shell-and-tube heat exchangers, unmeasured disturbances such as fluctuating feed composition and spatially coupled hydraulic–thermal waves in district-heating pipelines illustrate the resulting modeling challenges [3,4].

Classical first-principles models rigorously enforce conservation laws and constitutive relations, but often become intractable when geometries are complex or key parameters are uncertain. Conversely, purely data-driven models may provide accurate one-step predictions within the training domain, but typically require large labeled datasets and extrapolate poorly [5]. Over the last decade, hybrid or semi-mechanistic approaches have emerged that combine mechanistic insight with machine-learning surrogates. Recent reviews cover physics-informed neural networks (PINNs) for heat-transfer PDEs [4], gray-box network models for district heating [2] and knowledge-guided digital twins across chemical plants [6]. Collectively, these works demonstrate how prior information — mass and energy balances, constitutive equations, topology graphs, operating envelopes or monotonicity relations — can be embedded via constrained losses, differentiable simulators or structure-encoded architectures. Table 1 summarizes representative strategies for embedding such priors in machine-learning models of thermodynamic energy systems.

Despite this progress, most existing studies focus on one specific type of prior and design a bespoke learning objective around it. When several heterogeneous priors are available, three limitations become

* Corresponding author.
*E-mail address:* kummer.alex@mk.uni-pannon.hu (A. Kummer).

https://doi.org/10.1016/j.jprocont.2026.103785
Received 28 January 2026; Received in revised form 11 April 2026; Accepted 28 June 2026
Available online 9 July 2026

**Table 1**
Representative strategies for embedding prior information in machine-learning models of thermodynamic and process-industry energy systems.

| Method | Embedded prior information | Embedding technique | Target energy system | Ref. |
|---|---|---|---|---|
| Physics-Informed Neural Network (PINN) | Heat, mass and momentum PDEs; boundary conditions | PDE-residual terms added to composite loss | Two-phase heat exchanger | [7] |
| Decoupling–Coupling PINN | Mass/energy balances; reaction kinetics | Sequential subdomain pre-training followed by coupled physics loss | Plate chemical reactor | [8] |
| Physics-Informed RNN with network topology | Pipe topology graph; conservation laws | Physical graph hard-wired in RNN architecture | District-heating network | [9] |
| Gray-box state-space model | Energy-balance ODE structure; selected parameters | Fixed parametric structure with data-driven parameter estimation | Pilot heating process | [10] |
| Monotonicity-constrained PINN | Monotonic $NO_x$–fuel-input relationship | Inequality penalties enforcing monotonic outputs | Coal-fired boiler | [11] |
| PINN for borehole TES | Transient conduction PDE; flux boundary conditions | PDE residual plus initial/boundary loss terms | Borehole thermal-energy storage | [12] |
| PINN for heat-pump load | First-law energy balance; enthalpy relations | Energy-conservation term in loss | Absorption/geothermal heat pump | [13] |

apparent: (i) each knowledge source usually comes with its own hand-tuned weight in a composite loss; (ii) residuals from algebraic constraints, inequality heuristics and derivative matching live on different numerical scales, which makes trade-offs brittle; and (iii) fixed weighted sums cannot adapt when operating conditions or data quality change over time. As a consequence, combining multiple priors with data in a single, robust training objective remains challenging.

In this work we address these issues with the Utility-balanced Prior-informed Model (UPriMo), a utility-based framework for training surrogate models with heterogeneous prior knowledge. UPriMo maps the residual of each prior to a bounded utility and minimizes data misfit together with the shortfall from perfect utility, yielding scale-free and interpretable trade-offs without manual tuning of loss weights. Because the construction is agnostic to the underlying learner, it can be combined with state-space models, neural networks or differentiable simulators and acts as a plug-and-play bridge between mechanistic insight and data-driven flexibility. We demonstrate the framework on two canonical thermal-energy problems — a laminar pipe-flow surrogate and an electric water-heater model — and show that UPriMo improves accuracy, noise robustness and physical plausibility compared with purely data-driven baselines. The same recipe can be transferred to other energy-system modeling tasks.

Our contributions are threefold. First, we formulate UPriMo, a scale-free utility-based training objective that automatically balances data fidelity against multiple heterogeneous priors and supports physics-informed surrogate modeling. Second, we demonstrate the framework on two canonical problems — laminar pipe flow and electric water-heater dynamics — achieving an order-of-magnitude improvement in accuracy and robustness over purely data-driven baselines. Third, we provide a practical recipe for incorporating algebraic constraints, differential operators, monotonicity rules and surrogate-model references within a single training loop and standard machine-learning frameworks.

The remainder of this paper is organized as follows. Section 2 details the utility-based formulation and the incorporation of available prior knowledge. Section 3 describes the two case studies and experimental protocols and discusses the main results and ablation analyses. Finally, Section 4 concludes the paper and outlines directions toward multiscale industrial deployment.

## 2. Utility-function framework for integrating prior knowledge

We consider a supervised modeling task in which an unknown nonlinear mapping $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ relates independent variables $\mathbf{x} \in \mathbb{R}^{n_x}$ to outputs $\mathbf{y} \in \mathbb{R}^{n_y}$. Our goal is to learn an approximation

$$\hat{\mathbf{y}} = f(\mathbf{x}; \boldsymbol{\theta}),$$

where $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ is a parametric nonlinear model, typically a neural network whose trainable weights and biases are collected in $\boldsymbol{\theta} \in \mathbb{R}^p$. In modern machine-learning architectures the number of tunable parameters is often comparable to or larger than the number of available *labeled* measurements ($p \sim 10^3$–$10^5$ versus $N_m$ in typical engineering datasets). While acquiring additional labels can be prohibitively expensive, many engineering domains provide abundant *a priori* knowledge in analytic, heuristic or physics-based form. The utility-function framework introduced in this section exploits such prior knowledge to regularize the learning problem. Throughout, we assume that $f$ is differentiable with respect to both its inputs and parameters, a condition satisfied by standard neural networks and other differentiable surrogate models.

### 2.1. Problem definition and notation

We arrange the independent variables into two matrices, allowing a compact multiple-input multiple-output (MIMO) formulation. Let $\mathbf{x}_k^{(m)} \in \mathbb{R}^{n_x}$ and $\mathbf{y}_k^{(m)} \in \mathbb{R}^{n_y}$ denote the $k$th input and output column vectors for $k = 1, \dots, N_m$, and let $\mathbf{x}_\ell^{(c)} \in \mathbb{R}^{n_x}$ denote collocation inputs for $\ell = 1, \dots, N_c$. We then define

$$\mathbf{X}^{(m)} = \begin{bmatrix} (\mathbf{x}_1^{(m)})^\top \\ \vdots \\ (\mathbf{x}_{N_m}^{(m)})^\top \end{bmatrix} \in \mathbb{R}^{N_m \times n_x}, \quad \mathbf{Y}^{(m)} = \begin{bmatrix} (\mathbf{y}_1^{(m)})^\top \\ \vdots \\ (\mathbf{y}_{N_m}^{(m)})^\top \end{bmatrix} \in \mathbb{R}^{N_m \times n_y}, \tag{2.1}$$

$$\mathbf{X}^{(c)} = \begin{bmatrix} (\mathbf{x}_1^{(c)})^\top \\ \vdots \\ (\mathbf{x}_{N_c}^{(c)})^\top \end{bmatrix} \in \mathbb{R}^{N_c \times n_x}. \tag{2.2}$$

The upper indices $(m)$ and $(c)$ indicate *measurement* and *collocation*, respectively, while the lower indices $k = 1, \dots, N_m$ and $\ell = 1, \dots, N_c$ enumerate individual samples. Accordingly, the $k$th row of $\mathbf{X}^{(m)}$ (resp. the $\ell$th row of $\mathbf{X}^{(c)}$) is the row vector $(\mathbf{x}_k^{(m)})^\top$ (resp. $(\mathbf{x}_\ell^{(c)})^\top$).

When paired input–output observations are available, the learning task reduces to conventional nonlinear regression and is modeled as

$$\mathbf{y}_k^{(m)} = f\big(\mathbf{x}_k^{(m)}; \boldsymbol{\theta}\big) + \boldsymbol{\varepsilon}_k^{(m)}, \qquad k = 1, \dots, N_m, \tag{2.3}$$

where $\boldsymbol{\varepsilon}_k^{(m)}$ captures model mismatch and the prediction of the model is denoted $\hat{\mathbf{y}}_k^{(m)} = f\big(\mathbf{x}_k^{(m)}; \boldsymbol{\theta}\big)$. The parameter optimization task is to minimize the deviation from the measured outputs $\mathbf{y}_k^{(m)}$.

A standard choice for fitting $\boldsymbol{\theta}$ is the quadratic data loss

$$\mathcal{L}_{\text{data}}(\boldsymbol{\theta}) = \frac{1}{N_m} \sum_{k=1}^{N_m} \big(\mathbf{y}_k^{(m)} - \hat{\mathbf{y}}_k^{(m)}\big)^\top \mathbf{Q} \, \big(\mathbf{y}_k^{(m)} - \hat{\mathbf{y}}_k^{(m)}\big), \tag{2.4}$$

with $\mathbf{Q} \in \mathbb{R}^{n_y \times n_y}$ symmetric positive definite (e.g. the identity matrix or an empirical noise-covariance estimate).

Evaluating the model at new input locations $\mathbf{x}$ is typically cheap, so the input space $\mathbb{R}^{n_x}$—and its dense sampling matrix $\mathbf{X}^{(c)} \in \mathbb{R}^{N_c \times n_x}$ with

$N_c \gg N_m$—can be explored without acquiring new labels. In contrast, each labeled output $\mathbf{y}$ is costly to obtain. Consequently $N_m$ is often orders of magnitude smaller than what would be required for reliable coverage of the domain. A purely data-driven objective such as Eq. (2.4) therefore suffers from:

- *overfitting risk* — the model may memorize noise present in the scarce labels;
- *poor generalization* — sparse labels offer little guidance in unexplored regions of the input space;
- *sensitivity to noise and sampling bias*.

These shortcomings motivate augmenting $\mathcal{L}_{\text{data}}$ with additional constraint-based loss terms derived from available domain-specific knowledge. This additional structure guides the model toward more robust and generalizable solutions, even in sparsely measured regions of the input space.

### 2.2. Embedding prior information via collocation points

A common strategy is to embed prior information directly into the training objective. Physics- or model-based terms can be evaluated at arbitrarily many input locations $\mathbf{x}_\ell^{(c)}$, without additional $\mathbf{y}$ measurements, thereby increasing the effective information available for training. In physics-informed neural networks (PINNs) [14–16], spatial and/or temporal derivatives of $\hat{\mathbf{y}}$ are constrained to satisfy the governing partial differential equations (PDEs). The same idea extends well beyond PDEs: virtually any analytic or heuristic engineering knowledge—such as monotonicity, passivity, equality or inequality constraints and reduced-order surrogate models—can serve as a prior. Other examples include impulse/step-response matching, inverse constraints and derivative-based regularization.

Having established that any differentiable learner can be queried efficiently at collocation points, we can translate the full spectrum of prior engineering knowledge into machine-interpretable constraints. Each constraint is expressed through a residual function evaluated at the collocation inputs; these residuals then enter the composite training objective. In UPriMo, each prior loss is subsequently mapped to a bounded utility (Section 2.5).

### 2.3. Constraint formulation and prior-based loss functions

We translate each piece of prior information into a nonnegative *residual* $r_i \geq 0$. This residual quantifies the violation of the $i$th constraint and equals zero if and only if the constraint is fully satisfied:

$$r_i\Big(f(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta}),\, \nabla f(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta});\, \mathbf{x}_\ell^{(c)}\Big) \;=\; 0,$$

where $f(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta})$ is the model prediction evaluated at the collocation input $\mathbf{x}_\ell^{(c)}$. The corresponding prior loss is the mean residual over all collocation points:

$$\mathcal{L}_i^{\text{prior}}(\boldsymbol{\theta}) = \frac{1}{N_c}\sum_{\ell=1}^{N_c} r_i\Big(f(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta}),\, \nabla f(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta});\, \mathbf{x}_\ell^{(c)}\Big). \tag{2.5}$$

Typical residuals $r_i(\cdot)$ used in this work include reference-model fits, PDE residuals, derivative matching, monotonicity constraints, operational limits and inverse consistency. Section 2.4 details how each of these terms instantiates the generic template above.

The overall training objective is then

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \sum_{i=1}^{N_p} \mathcal{L}_i^{\text{prior}}(\boldsymbol{\theta}). \tag{2.6}$$

The prior knowledge-based constraints are evaluated at densely sampled collocation points as an extended set of inputs. Two conceptually distinct collocation sets are useful in practice:

(i) **Constraint-only points.** For priors based on derivative constraints, PDE residuals, monotonicity, convexity, or general equality/inequality conditions, no target output exists at $\mathbf{x}_\ell^{(c)}$. The model prediction $\hat{\mathbf{y}}_\ell^{(c)} := f(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta})$ is simply required to satisfy the chosen constraints.

(ii) **Reference- or surrogate-model points.** When a reference or reduced-order model exists (e.g., a state-space representation, impulse/step responses or an input–output surrogate), its inputs are treated as collocation points. A fixed-parameter model $f^{\text{R}}(\cdot;\boldsymbol{\theta}^{\text{R}})$ (identified beforehand) provides target outputs $\tilde{\mathbf{y}}_\ell^{(c)} := f^{\text{R}}\big(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta}^{\text{R}}\big)$, and the learner is trained to match these targets even though the pairs never appear in the measured dataset.

Both types of collocation points enrich the training objective with domain-specific constraints, improving predictive accuracy and robustness without requiring additional measured outputs. We next describe how prior information is translated into concrete residuals and penalties.

### 2.4. From classical regularization to knowledge-driven constraints

When *no* explicit prior information is available, the learning problem reverts to classical *weight-decay* regularization. In its isotropic variant, a ridge term

$$\mathcal{L}_{\text{Ridge}}(\boldsymbol{\theta}) = \lambda_{\text{ridge}}\|\boldsymbol{\theta}\|_2^2, \quad \lambda_{\text{ridge}} > 0 \tag{2.7}$$

augments the data loss (2.4), thereby preventing unbounded parameter growth and mitigating overfitting. The ridge penalty can thus be viewed as the baseline member of a much richer family of *constraint-induced* residuals described below. The knowledge-based residuals operate on the dense collocation set $\mathbf{X}^{(c)}$ and enter the total objective together with the data loss. In what follows, we catalogue the most frequently encountered examples.

For clarity, all residuals are formulated for an arbitrary collocation input $\mathbf{x}_\ell^{(c)}$; aggregation over $\ell = 1,\dots,N_c$ is implicit and follows the pattern of Eq. (2.5). Where relevant, we explicitly indicate whether a residual is used at constraint-only or reference-model collocation points as defined in Section 2.3.

*1. Reference-model consistency.* Let $\tilde{\mathbf{y}}_\ell^{(c)} := f^{\text{R}}\big(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta}^{\text{R}}\big)$ be the prediction of a reduced-order or physics-based reference with *fixed* parameters $\boldsymbol{\theta}^{\text{R}}$. Penalizing pointwise deviations yields

$$\mathcal{L}_{\text{Reference}}^{\text{prior}} = \frac{1}{N_c}\sum_{\ell=1}^{N_c}\Big\|\hat{\mathbf{y}}_\ell^{(c)} - \tilde{\mathbf{y}}_\ell^{(c)}\Big\|_2^2. \tag{2.8}$$

This residual is the multi-output analog of classical model-following control.

*2. Derivative-based priors.* For brevity, $\partial^q\hat{\mathbf{y}}_\ell^{(c)}$ denotes the tensor of $q$th order partial derivatives for each output component with respect to each input dimension:

$$\partial^q\hat{\mathbf{y}}_\ell^{(c)} := \frac{\partial^q\hat{\mathbf{y}}_\ell^{(c)}}{\partial\mathbf{x}_\ell^{(c)\,q}}, \qquad \partial^q\tilde{\mathbf{y}}_\ell^{(c)} := \frac{\partial^q f^{\text{R}}\big(\mathbf{x}_\ell^{(c)};\boldsymbol{\theta}^{\text{R}}\big)}{\partial\mathbf{x}_\ell^{(c)\,q}}. \tag{2.9}$$

Three generic residual types are common.

(a) *PDE residual.* Suppose a differential operator $\mathcal{F}\big(\mathbf{x}_\ell^{(c)},\hat{\mathbf{y}}_\ell^{(c)},\partial^q\hat{\mathbf{y}}_\ell^{(c)},\boldsymbol{\theta}^{\text{M}}\big) = \mathbf{0}$ expresses the governing physics. Its violation is penalized by

$$\mathcal{L}_{\text{PDE}}^{\text{prior}} = \frac{1}{N_c}\sum_{\ell=1}^{N_c}\Big\|\mathcal{F}\big(\mathbf{x}_\ell^{(c)},\hat{\mathbf{y}}_\ell^{(c)},\partial^q\hat{\mathbf{y}}_\ell^{(c)},\boldsymbol{\theta}^{\text{M}}\big)\Big\|_2^2. \tag{2.10}$$

Here, $\boldsymbol{\theta}^{\text{M}}$ refers to parameters in the governing equation; if unknown, they are estimated jointly with $\boldsymbol{\theta}$.

(b) *Derivative matching to a reference model.* If a surrogate model $f^{\mathrm{R}}(\cdot;\theta^{\mathrm{R}})$ is available, one may impose agreement of the $q$th derivatives:

$$\mathcal{L}^{\text{prior}}_{\text{Derivative}} = \frac{1}{N_c}\sum_{\ell=1}^{N_c}\left\|\partial^q \hat{\mathbf{y}}^{(c)}_\ell - \partial^q \tilde{\mathbf{y}}^{(c)}_\ell\right\|_2^2. \tag{2.11}$$

(c) *Sign-constrained derivatives (monotonicity/convexity).* Let $d \in \{1,\dots,n_x\}$ denote the $d$th input coordinate of the row vector $\mathbf{x}^{(c)}_\ell = [x^{(c)}_{\ell,1},\dots,x^{(c)}_{\ell,n_x}]$, and let $j \in \{1,\dots,n_y\}$ denote the output dimension. If domain knowledge prescribes monotonicity constraints for the $j$th output component with respect to the $d$th input dimension, we introduce the following one-sided ReLU-type residual:

$$\mathcal{L}^{\text{prior}}_{\text{Sign}} = \frac{1}{N_c}\sum_{\ell=1}^{N_c}\max\left\{0,\ -s_{dj}\,\frac{\partial \hat{y}^{(c)}_{\ell,j}}{\partial x^{(c)}_{\ell,d}}\right\}, \tag{2.12}$$

where $s_{dj} \in \{+1,-1\}$ encodes the desired monotonicity direction: $+1$ corresponds to monotonically increasing and $-1$ to monotonically decreasing. The penalty activates only when the prescribed sign constraint is violated, thereby enforcing monotonicity without altering regions that already comply.

*3. Heuristic inequality constraints.* Given a scalar performance index $g : \mathbb{R}^{n_y} \to \mathbb{R}$ computed from the model output, we may want to enforce either $g(f(\mathbf{x}^{(c)}_\ell;\theta)) \le \varepsilon$ (upper bound) or $g(f(\mathbf{x}^{(c)}_\ell;\theta)) \ge \varepsilon$ (lower bound). Both cases can be written compactly with a sign parameter $s_g \in \{+1,-1\}$:

$$\mathcal{L}^{\text{prior}}_{\text{Inequality}} = \frac{1}{N_c}\sum_{\ell=1}^{N_c}\left[\max\left\{0,\ s_g\left(g(\hat{\mathbf{y}}^{(c)}_\ell) - \varepsilon\right)\right\}\right]^2, \tag{2.13}$$

where

$$s_g = \begin{cases} +1, & \text{if an } \textit{upper} \text{ bound } g(\cdot) \le \varepsilon \text{ is required,} \\ -1, & \text{if a } \textit{lower} \text{ bound } g(\cdot) \ge \varepsilon \text{ is required.} \end{cases}$$

The hinge loss activates only when the inequality is violated, softly pushing predictions back into the admissible region.

*4. Inverse-mapping residual.* If $f(\cdot;\theta)$ is invertible, one can enforce that the estimated inverse recovers the input:

$$\mathcal{L}^{\text{prior}}_{\text{Inverse}} = \frac{1}{N_c}\sum_{\ell=1}^{N_c}\left\|f^{-1}(\hat{\mathbf{y}}^{(c)}_\ell;\theta) - \mathbf{x}^{(c)}_\ell\right\|_2^2. \tag{2.14}$$

With these residuals integrated into (2.6), the optimization jointly exploits measured data and domain knowledge, yielding models that generalize better than purely data-driven counterparts.

### 2.5. Utility-based normalization of prior residuals

Because prior losses typically vary greatly in magnitude and units, direct summation in (2.6) can lead to numerical imbalances and makes the optimization sensitive to arbitrary scaling. To mitigate this, we define a bounded utility function $U_i : [0,\infty) \to [0,1]$ that is monotonically decreasing with respect to the prior loss $\mathcal{L}^{\text{prior}}_i$.

As $\mathcal{L}^{\text{prior}}_i$ approaches zero (perfect satisfaction), the utility approaches one, whereas for large violations it tends to zero. This property yields a normalized, dimensionless measure of prior satisfaction and helps balance heterogeneous constraints without explicit weighting.

Typical choices include

- *rational:* $U_i(\mathcal{L}^{\text{prior}}_i) = \left(1+\alpha_i\,\mathcal{L}^{\text{prior}}_i\right)^{-1}$;
- *exponential:* $U_i(\mathcal{L}^{\text{prior}}_i) = \exp\left(-\alpha_i\,\mathcal{L}^{\text{prior}}_i\right)$;
- *Gaussian (Taguchi-like):* $U_i(\mathcal{L}^{\text{prior}}_i) = \exp\left(-(\alpha_i\,\mathcal{L}^{\text{prior}}_i)^2\right)$;
- *ReLU-based saturation:* $U_i(\mathcal{L}^{\text{prior}}_i) = \max\{0,\ 1-\alpha_i\,\mathcal{L}^{\text{prior}}_i\}$.

Here $\alpha_i > 0$ is a shape parameter chosen to achieve the desired sensitivity. A constraint with a large residual then yields a small utility, whereas a nearly satisfied constraint yields a utility close to one. With this normalization each prior contributes the bounded *utility shortfall* term $1-U_i \in [0,1]$ to the objective, which facilitates comparison with the data loss even when the underlying physical units differ.

Replacing each prior residual term by its utility shortfall, the parameter optimization problem can be written as

$$\hat{\theta},\ \hat{\theta}^{\mathrm{M}} = \arg\min_{\theta,\theta^{\mathrm{M}}}\left[\mathcal{L}_{\text{data}}(\theta) + \sum_{i=1}^{N_p}\left(1-U_i\left(\mathcal{L}^{\text{prior}}_i(\theta)\right)\right)\right], \tag{2.15}$$

where the tunable parameter vector $\theta$ may be augmented by $\theta^{\mathrm{M}}$ (unknown physical parameters; see Eq. (2.10)). Because $0 \le 1-U_i \le 1$ for all $i$, every prior term is automatically normalized and bounded, while the interpretation remains intuitive: $U_i \to 1$ implies near-perfect satisfaction of the $i$th constraint, whereas $U_i \to 0$ signals significant violation.

The proposed formulation offers three major advantages for parameter optimization and constraint balancing:

- **Bounded influence.** No single prior can dominate the optimization, even if its raw residual is large, which helps to prevent numerical imbalance.
- **Scale-free tuning.** The shape parameters $\alpha_i$ replace ad hoc weights $\lambda_i$; they control how sharply the utility drops but have a fixed output range, simplifying hyperparameter selection.
- **Unified stopping criterion.** Since all utilities lie in $[0,1]$, one may, for instance, require that the average utility exceed a designated threshold before terminating training.

The utility-based objective in Eq. (2.15) can be interpreted as a nonlinear scalarization of the data loss and the individual prior residuals, and therefore as a particular multi-objective formulation. In this view, UPriMo does not introduce a separate optimization class, but applies a bounded transformation to each prior term before aggregation. Compared with direct residual summation, this changes the optimization behavior by limiting the influence of very large prior violations and making heterogeneous priors more comparable. When priors are conflicting, the method does not resolve the conflict explicitly; instead, it seeks a compromise solution in which the individual objectives may be satisfied to different degrees. Thus, the obtained solution can be interpreted as a scalarization-induced Pareto compromise, rather than the simultaneous exact satisfaction of all priors. Because the resulting training problem remains nonconvex in the neural-network setting considered here, we do not claim global convergence guarantees. Our contribution is instead to clarify that the bounded utility transformation modifies the trade-off structure of the optimization in a controlled and interpretable way.

Unlike adaptive loss-balancing approaches (e.g., gradient-based reweighting or uncertainty-based weighting), which operate by dynamically adjusting scalar coefficients in a weighted sum of heterogeneous loss terms, UPriMo modifies the functional form of each prior contribution before aggregation. Each prior residual is mapped to a bounded utility shortfall on a common [0,1] scale, which makes heterogeneous priors directly comparable and prevents any single prior from exerting unbounded influence on the aggregated prior contribution. This transformation can be interpreted as a nonlinear scalarization of multiple objectives, rather than a linear combination with adaptive weights. The contribution is therefore not the removal of all hyperparameter choices, but the introduction of a common bounded utility-based normalization that replaces ad hoc cross-scale balancing with a more uniform sensitivity specification.

The forthcoming application sections (Section 3) will demonstrate how the utility-based formulation integrates derivative, heuristic and reference-model priors within a single optimization loop, thereby improving robustness and generalization.

**Table 2**
Laminar pipe-flow parameters.

| Parameter | Definition | Value and SI unit |
|---|---|---|
| $\rho$ | Fluid density | 997 [kg m$^{-3}$] |
| $\mu$ | Dynamic viscosity | $1.002 \times 10^{-3}$ [Pa s] |
| $R$ | Pipe radius | 1.0 [m] |
| $Q$ | Volumetric flow rate | $5.0 \times 10^{-4}$ [m$^3$ s$^{-1}$] |
| $G$ | Pressure gradient | $-\frac{8\mu Q}{\pi R^4}$ [Pa m$^{-1}$] |

## 3. Case studies and applications

To illustrate the proposed framework in realistic settings, we consider two representative energy-system models. The first is fully developed laminar pipe flow, a steady fluid-dynamics benchmark with an analytic Hagen–Poiseuille solution. This setting allows precise assessment of interpolation and extrapolation performance under extremely sparse and noisy data. The second is an electric water heater, a nonlinear dynamical system modeled in NARX form, where long-horizon free-run simulations amplify modeling errors unless constrained by suitable priors. Together, these examples demonstrate how diverse prior knowledge — from governing equations and boundary conditions to monotonicity rules and steady-state relations — can be encoded through the utility-based formulation of Section 2 to improve accuracy and robustness over purely data-driven baselines.

### 3.1. Laminar pipe flow: a sparse and noisy benchmark for physics-informed nonlinear regression

Fully developed laminar flow in a circular pipe is a standard benchmark with a known parabolic velocity profile, providing an absolute error reference. When only a handful of noisy samples are available, however, the regression task becomes ill-posed. Reconstructing the axial velocity $u(r)$ from five perturbed points inside the inner half-radius region ($\tilde{r} \in [-R/2, R/2]$) forces the learner to interpolate between sparse measurements and, more critically, to extrapolate toward the wall. This setting mirrors practical scenarios where dense field data are prohibitive but wall-shear or pressure-loss estimates remain essential, making laminar pipe flow an ideal testbed for physics-informed nonlinear regression.

We adopt a right-handed cylindrical coordinate system $(r, \theta, z)$, with the $z$-axis aligned with the pipe centerline and $r$ measuring the radial distance from this axis. The working fluid is incompressible and Newtonian, with constant density $\rho$ [kg m$^{-3}$] and dynamic viscosity $\mu$ [Pa s], driven by a constant (negative) axial pressure gradient

$$G := \frac{dp}{dz} \ [\text{Pa m}^{-1}],$$

under laminar conditions corresponding to Reynolds numbers Re $<$ 2300. For reference,

$$\text{Re} = \frac{\rho \bar{u} D}{\mu}, \qquad \bar{u} = \frac{Q}{\pi R^2}, \qquad D = 2R,$$

where $\bar{u}$ denotes the cross-sectional average velocity, $Q$ the volumetric flow rate and $R$ the pipe radius. Under the assumptions of steady, fully developed, unidirectional flow the velocity field $\mathbf{u}(r, \theta, z) = (u_r, u_\theta, u_z)$ reduces to

$$(u_r, u_\theta, u_z) = (0, 0, u(r)),$$

so only the axial component is nonzero and depends solely on $r$. For numerical convenience we introduce a signed radial coordinate $\tilde{r} \in [-R, R]$ related to the physical radius by $r = |\tilde{r}|$. Axisymmetry implies $u(\tilde{r}) = u(-\tilde{r})$, so sampling across both positive and negative radii effectively doubles spatial resolution near the pipe centerline without introducing new physics. The fluid-dynamical parameters used throughout the study are summarized in Table 2.

Under these assumptions, the steady incompressible Navier–Stokes equations reduce to the classical Hagen–Poiseuille ordinary differential equation

$$\mu \frac{1}{\tilde{r}} \frac{d}{d\tilde{r}} \left( \tilde{r} \frac{du}{d\tilde{r}} \right) = G, \qquad G = -\frac{8\mu Q}{\pi R^4}, \tag{3.1}$$

whose closed-form solution is

$$u^\star(\tilde{r}) = \frac{2Q}{\pi R^2} \left( 1 - \frac{\tilde{r}^2}{R^2} \right), \qquad |\tilde{r}| \le R. \tag{3.2}$$

We use $u^\star(\tilde{r})$ as ground truth when computing validation errors; for a prescribed flow rate $Q$ the profile is independent of viscosity.

From a nonlinear regression viewpoint, the learner maps radial positions to predicted velocities as

$$\hat{u} = f(\tilde{r}; \boldsymbol{\theta}), \qquad f : \mathbb{R} \to \mathbb{R}, \tag{3.3}$$

where $\boldsymbol{\theta}$ denotes the network parameters. Two input sets are defined. The *sparse data points* are $\{\mathbf{x}_k^{(m)} = \tilde{r}_k\}_{k=1}^{N_m}$, with $N_m = 5$ uniformly sampled from $[-R/2, R/2]$; their noisy targets are

$$y_k = u^\star(\tilde{r}_k) + \varepsilon_k, \qquad \varepsilon_k \sim \mathcal{N}(0, \sigma^2).$$

The *collocation points* are $\{\mathbf{x}_\ell^{(c)} = \tilde{r}_\ell\}_{\ell=1}^{N_c}$, with $N_c = 30$ (quasi-)uniformly distributed over $[-R, R]$ excluding the center point $\tilde{r} = 0$ due to the $1/\tilde{r}$ operator in the PDE residual; the centerline condition is handled separately via a derivative prior at $\tilde{r} = 0$. Collocation points have no target outputs and serve solely to evaluate prior residuals (PDE, boundary, monotonicity, etc.). The combined objective therefore includes the data term and five prior utilities ($U_{\text{PDE}}$, $U_{\text{BC}}$, $U_{\text{Mono}}$, $U_{\text{Max}}$, $U_{\text{Re}}$) defined by the utility-based formulation in Section 2.5, with the PDE residual normalized by $\kappa = 8Q/(\pi R^4)$ as discussed in (P1) to ensure scale compatibility across priors.

We incorporate the following domain-specific priors for laminar flow:

**(P1) PDE loss.** The Navier–Stokes residual for fully developed laminar pipe flow is defined pointwise as

$$r_{\text{PDE}}(\tilde{r}_\ell) = \frac{1}{\kappa} \left[ \frac{1}{\tilde{r}_\ell} \frac{\partial}{\partial \tilde{r}_\ell} \left( \tilde{r}_\ell \frac{\partial \hat{u}}{\partial \tilde{r}_\ell} \right) + \kappa \right], \qquad \kappa = \frac{8Q}{\pi R^4}. \tag{3.4}$$

The corresponding normalized prior loss is the mean of squared residuals

$$\mathcal{L}_{\text{PDE}}^{\text{norm}} = \frac{1}{N_c} \sum_{\ell=1}^{N_c} r_{\text{PDE}}(\tilde{r}_\ell)^2. \tag{3.5}$$

Scaling by $\kappa$ yields a dimensionless residual and keeps $\mathcal{L}_{\text{PDE}}^{\text{norm}}$ on the order of unity, largely independent of the flow rate $Q$ and pipe radius $R$. This normalization improves numerical stability and ensures compatibility with the bounded-utility formulation of Section 2.5. In practice we exclude the centerline $\tilde{r} = 0$ from PDE collocation, since the operator contains $1/\tilde{r}$, and enforce regularity at the axis via (P4).

**(P2) Boundary condition (no-slip).** The no-slip condition is a Dirichlet-type boundary constraint stating that a viscous fluid must have zero velocity relative to a solid wall. In our cylindrical coordinates this implies

$$u(\tilde{r}_\ell = \pm R) = 0.$$

We encode this prior through the scalar loss

$$\mathcal{L}_{\text{BC}} = \tfrac{1}{2} \left[ \hat{u}(R)^2 + \hat{u}(-R)^2 \right], \tag{3.6}$$

which penalizes any deviation of the predicted wall velocity from zero, thereby enforcing the physical no-slip behavior without introducing additional penalty weights.

**(P3) Monotonicity constraint.** A radially decreasing velocity profile is enforced via a two-sided ReLU-based penalty:

$$\mathcal{L}_{\text{Mono}} = \max\left\{0,\ \max_{\tilde{r}_\ell \geq 0} \partial_{\tilde{r}_\ell} \hat{u}(\tilde{r}_\ell)\right\} + \max\left\{0,\ -\min_{\tilde{r}_\ell < 0} \partial_{\tilde{r}_\ell} \hat{u}(\tilde{r}_\ell)\right\}. \quad (3.7)$$

This formulation separately penalizes positive radial velocity gradients on the positive side ($\tilde{r} > 0$) and negative gradients on the negative side ($\tilde{r} < 0$). The ReLU function ensures a sparse penalty that activates only when monotonicity is violated.

**(P4) Centerline maximum velocity.** The maximum axial velocity occurs at the pipe axis ($\tilde{r} = 0$). Consequently, its first radial derivative must vanish at that location:

$$\mathcal{L}_{\text{Max}} = \left[\partial_{\tilde{r}_\ell} \hat{u}(\tilde{r}_\ell)\right]^2\Big|_{\tilde{r}_\ell = 0}. \quad (3.8)$$

This regularity condition complements (P1) at the excluded centerline point.

**(P5) Reynolds-number utility.** Laminar flow conditions are encouraged directly through a logistic utility computed from the predicted maximum axial velocity. Let

$$u_{\max} := \max_{\ell} \hat{u}(\tilde{r}_\ell), \qquad \text{Re}_{u_{\max}} = \frac{\rho\, u_{\max}\, D}{\mu}.$$

We define

$$U_{\text{Re}} = \frac{1}{1 + \exp\left[(\text{Re}_{u_{\max}} - \text{Re}_{\max})/\tau_{\text{Re}}\right]}, \qquad \text{Re}_{\max} = 2300, \quad \tau_{\text{Re}} = 0.01, \quad (3.9)$$

where $\tau_{\text{Re}}$ controls the transition width and $\text{Re}_{\max}$ marks the classical laminar–turbulent threshold. The threshold refers to an average-velocity Reynolds number; using $u_{\max}$ is conservative, since for a parabolic profile $u_{\max} = 2\bar{u}$. Unlike the previous priors, $U_{\text{Re}}$ directly serves as a utility without an explicit residual or loss term.

For (P1) the loss is obtained as a mean-squared residual over collocation points. For (P2)–(P4) we directly define scalar nonnegative losses $\mathcal{L}_{\text{BC}}$, $\mathcal{L}_{\text{Mono}}$ and $\mathcal{L}_{\text{Max}}$ as in Eqs. (3.6)–(3.8). In all cases we obtain a prior loss $\mathcal{L}_i \geq 0$, which is mapped to a utility via the common rational transform

$$\boxed{U_i(\mathcal{L}_i) = \frac{1}{1 + \alpha\, \mathcal{L}_i}, \qquad \alpha = 100} \quad (3.10)$$

for $i \in \{\text{PDE}, \text{BC}, \text{Mono}, \text{Max}\}$. The mapping is monotonically decreasing, bounded in $(0, 1]$ and approaches one only for very small residuals ($\mathcal{L}_i \ll \alpha^{-1}$). Using a single $\alpha$ for (P1)–(P4) simplifies, though does not eliminate, hyperparameter tuning.

Combining the data term with the five utilities yields the composite objective

$$\mathcal{J}(\boldsymbol{\theta}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \left[1 - U_{\text{PDE}}(\mathcal{L}_{\text{PDE}}^{\text{norm}})\right] + \left[1 - U_{\text{BC}}(\mathcal{L}_{\text{BC}})\right] + \left[1 - U_{\text{Mono}}(\mathcal{L}_{\text{Mono}})\right] + \left[1 - U_{\text{Max}}(\mathcal{L}_{\text{Max}})\right] + \left[1 - U_{\text{Re}}\right], \quad (3.11)$$

which is minimized with respect to the network parameters $\boldsymbol{\theta}$. Each bracketed term lies in $[0, 1]$, ensuring that no single prior can dominate the optimization even when its raw residual is large.

Measurement points are sampled at five radial locations uniformly distributed within the interval $[-R/2, R/2]$ ($N_m = 5$). For each training instance we first normalize the *noiseless* reference velocities by the instance-wise peak

$$s := \max_{j \leq N_m} u^\star(\tilde{r}_j),$$

and define

$$\bar{y}_k = \frac{u^\star(\tilde{r}_k)}{s}.$$

**Table 3**
Experimental design for the laminar-flow study: factors and levels.

| Factor | Symbol | Levels |
|---|---|---|
| Noise standard deviation (normalized output units) | $\sigma$ | {0.00, 0.05, 0.10, 0.20} |
| Hidden layers | $L$ | {1, 2, 3} |
| Neurons per layer | $H$ | {5, 10, 15} |

Additive Gaussian noise is then applied in the normalized domain,

$$y_k = \bar{y}_k + \varepsilon_k, \qquad \varepsilon_k \sim \mathcal{N}(0, \sigma^2), \quad (3.12)$$

so that the noiseless targets lie in $[0, 1]$, while the noisy labels remain concentrated around this range for the noise levels considered. All data losses are computed on these normalized (and possibly noisy) targets.

*Important for physics-based priors.* Although the network is trained on normalized targets $y_k$, all physics-based priors (PDE, boundary, monotonicity, maximum, Reynolds) are evaluated on de-normalized predictions

$$\tilde{u}(\tilde{r}) = s\, \hat{u}(\tilde{r}),$$

so that the governing equations and utilities operate in physical units. For validation on the dense grid we use a consistent scale: the analytical reference $u^\star(\tilde{r})$ is normalized by the same factor $s$ before computing error metrics, ensuring that predictions and references are compared in matching units.

The experimental design follows a full factorial structure over noise level and network architecture; for each combination we train both a purely data-driven and a UPriMo-augmented model. The factors and their levels are summarized in Table 3.

Fully connected feedforward networks are used throughout. Architectures are indexed by the pair $(L, H)$, yielding nine configurations: $(1, 5)$, $(1, 10)$, $(1, 15)$, $(2, 5)$, $(2, 10)$, $(2, 15)$, $(3, 5)$, $(3, 10)$ and $(3, 15)$. For each noise level $\sigma$, each architecture $(L, H)$ and each training objective (baseline vs. UPriMo), we train a separate network, resulting in $4 \times 9 \times 2 = 72$ models in total for this case study, i.e., 36 runs per model type.

Both baselines and UPriMo models are trained under identical architectural and noise conditions; optimizer settings, batch size and random seeds are kept the same across methods for reproducibility. Models are evaluated using mean squared error (MSE), mean absolute error (MAE), root-mean-square error (RMSE) and coefficient of determination ($R^2$). For UPriMo, training stops when all utilities satisfy $U_i \geq 0.97$ or after 3000 epochs, whichever occurs first. Purely data-driven models use early stopping based on validation loss (patience = 10), monitored from epoch 400 onward. Validation is conducted on a dense radial grid (200 points uniformly spaced on $[-R, R]$), comparing predictions to the analytical reference (Eq. (3.2)) in the same normalization as training, i.e., the reference is scaled by the sample-wise factor $s$ defined in Eq. (3.12). For the Reynolds-number prior, the utility is computed from the de-normalized maximum velocity

$$u_{\max} = \max_{\tilde{r}} \tilde{u}(\tilde{r}) = s\, \max_{\tilde{r}} \hat{u}(\tilde{r}),$$

ensuring physical consistency with the utility-based formulation in Section 2.5.

For each combination of noise level and architecture in Table 3, we therefore obtain paired runs: a purely data-driven neural network and a UPriMo-augmented model trained on the same five noisy measurements. Generalization is assessed on 200 radial locations covering the full pipe cross-section, with the analytical Hagen–Poiseuille solution as ground truth. The four error metrics (MSE, MAE, RMSE and $R^2$) are monitored on both train and validation sets.

Table 4 reports the best validation RMSE at each noise level, aggregated over all architectures. Across the entire noise range, UPriMo outperforms the purely data-driven network by roughly two orders of magnitude: while the baseline RMSE increases from 0.169 to 0.254

**Table 4**
Best validation RMSE of each model at the four noise levels.

| Noise std. | Data-driven RMSE | UPriMo RMSE |
|---|---|---|
| 0.00 | 0.169 | **0.0033** |
| 0.05 | 0.178 | **0.0059** |
| 0.10 | 0.233 | **0.0033** |
| 0.20 | 0.254 | **0.0032** |

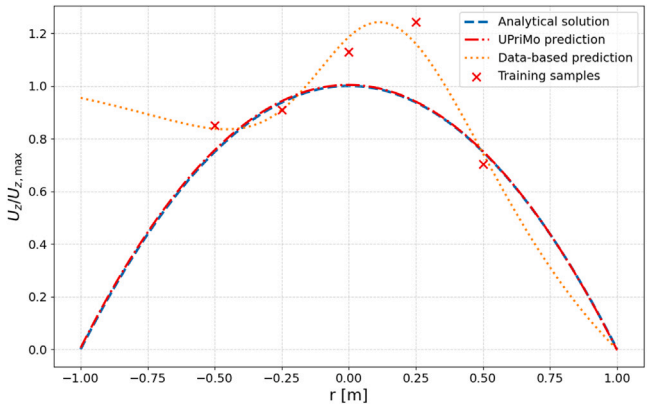


**Fig. 1.** Single-shot profile comparison at noise level $\sigma = 0.20$ (3 hidden layers, 10 neurons). The five training samples are marked by ×. UPriMo (red, solid) matches the analytical Poiseuille profile (blue, dashed) across the radius $r$ [m], while the data-only network (orange, dotted) exhibits a flattened apex and spurious near-wall behavior. Vertical axis: axial velocity $U_z/U_{z,max}$.

as $\sigma$ grows from 0 to 0.20, the UPriMo RMSE remains in the narrow interval $[0.0032, 0.0059]$. A similar pattern holds for the coefficient of determination: the data-driven $R^2$ deteriorates from about 0.62 to 0.23 with increasing noise, whereas UPriMo maintains $R^2 > 0.999$ in all cases.

The combination of PDE and derivative priors acts as a strong smoothness regularizer, while the Reynolds-limit and maximum-location utilities forbid implausible peaks or shape distortions. As a result, the UPriMo networks remain well-behaved even when trained on only five noisy observations, whereas the purely data-driven models become increasingly noise-sensitive.

A representative high-noise case is shown in Fig. 1 for $\sigma = 0.20$ and a network with three hidden layers and ten neurons per layer. The five training samples are marked by crosses. UPriMo (red, solid) closely matches the analytical Poiseuille profile (blue, dashed) across the full radius, while the data-only network (orange, dotted) exhibits a flattened apex and spurious near-wall behavior under the same training data. The prior utilities therefore not only lower error but also recover the correct physical *shape*: symmetry, zero wall velocity and maximal centerline speed.

To further isolate the effect of the proposed utility-based aggregation, we performed an additional controlled comparison on the laminar pipe-flow benchmark against a standard equal-weight PINN. The comparison was intentionally restricted to the same prior set already used in the laminar study — data fidelity, PDE residual, wall boundary condition, derivative-based regularity, and centerline maximum — while the Reynolds-number utility was omitted in this experiment. This yields a like-for-like comparison with a classical composite PINN objective, since both models are trained from the same measurements and collocation points and differ only in how the prior terms are aggregated.

More specifically, both models use the same network architecture, the same noisy training samples, the same collocation set, and the same optimizer settings. The equal-weight PINN minimizes the direct sum of the raw loss terms,

$$L_{\text{PINN}} = L_{\text{data}} + L_{\text{PDE}} + L_{\text{BC}} + L_{\text{Der}} + L_{\text{Max}}, \tag{3.13}$$

whereas UPriMo uses the same raw residuals but maps the prior terms through the bounded utility transform introduced in Section 2.5. In this way, the experiment isolates the contribution of the utility-based objective itself, rather than conflating it with different architectures or different prior definitions.

The comparison was carried out for a representative high-noise setting ($\sigma = 0.20$) and a fixed network architecture with three hidden layers and ten neurons per layer. To assess robustness with respect to initialization, each model was trained with three random seeds. Fig. 2 reports the final loss components, Fig. 3 summarizes the training-cost indicators, and Fig. 4 shows the validation metrics on the dense analytical reference grid.

The results show a consistent pattern across seeds. First, UPriMo reaches lower validation error than the equal-weight PINN, with visibly smaller RMSE and MAE and with $R^2$ values essentially indistinguishable from unity. Second, the final loss-component distributions indicate that the utility-based objective attains a more balanced compromise between data fit and prior satisfaction, rather than overemphasizing individual raw residual terms. This behavior is consistent with the bounded-shortfall construction in Eq. (2.15): no single prior can dominate the optimization solely because of its numerical scale.

The training-cost comparison is also informative. Since both models evaluate the same derivative-based and PDE-based residuals, their per-epoch computational workload remains of the same order. The observed reduction in wall-clock training time for UPriMo is therefore mainly attributable to faster convergence to an acceptable compromise solution, rather than to a fundamentally different automatic-differentiation complexity.

Overall, the laminar pipe-flow study shows that embedding a small set of domain constraints into the loss function yields (i) two orders of magnitude lower RMSE, (ii) graceful degradation under measurement noise and (iii) reduced sensitivity to architecture hyperparameters. In the sequel we challenge the framework with a less structured, non-stationary system — the electric water heater — where Reynolds-type priors are insufficient and temporal consistency must also be enforced.

### 3.2. NARX formulation - physics-informed modeling of an electrical heater

We represent the heater dynamics by a nonlinear discrete-time system whose current output depends on past outputs and delayed inputs, adopting a standard *nonlinear autoregressive with exogenous inputs* (NARX) structure widely used in nonlinear system identification [17]. The physical plant exhibits a process dead time due to transport and mixing, so changes in the inputs affect the outlet temperature only after a finite delay.

Let $U(t)$, $Q(t)$, and $T_{\text{out}}(t)$ denote the continuous-time signals. Under uniform sampling with period $T_s > 0$, define

$$u_k := U(kT_s), \qquad q_k := Q(kT_s), \qquad y_k := T_{\text{out}}(kT_s), \quad k \in \mathbb{Z}.$$

We model causal, strictly proper dynamics with integer input delays (dead times) measured in samples. For input delays $d_u, d_q \in \mathbb{N}$, the most recent input samples that can influence $y_k$ are $u_{k-1-d_u}$ and $q_{k-1-d_q}$, respectively.

Let $n_a, n_{b,u}, n_{b,q} \in \mathbb{N}$ denote the autoregressive and input orders. Define the histories

$$\mathbf{u}_{k-d_u} := \left[u_{k-1-d_u}, u_{k-2-d_u}, \ldots, u_{k-n_{b,u}-d_u}\right], \tag{3.14}$$

$$\mathbf{q}_{k-d_q} := \left[q_{k-1-d_q}, q_{k-2-d_q}, \ldots, q_{k-n_{b,q}-d_q}\right], \tag{3.15}$$

$$\mathbf{y}_k := \left[y_{k-1}, y_{k-2}, \ldots, y_{k-n_a}\right], \tag{3.16}$$

and stack them into the regressor

$$\mathbf{x}_k := \left[\mathbf{u}_{k-d_u}, \mathbf{q}_{k-d_q}, \mathbf{y}_k\right]^\top, \qquad k \geq 1+\max\{n_a, d_u+n_{b,u}, d_q+n_{b,q}\}. \tag{3.17}$$

By construction, the model is causal and strictly proper; inputs do not act instantaneously.

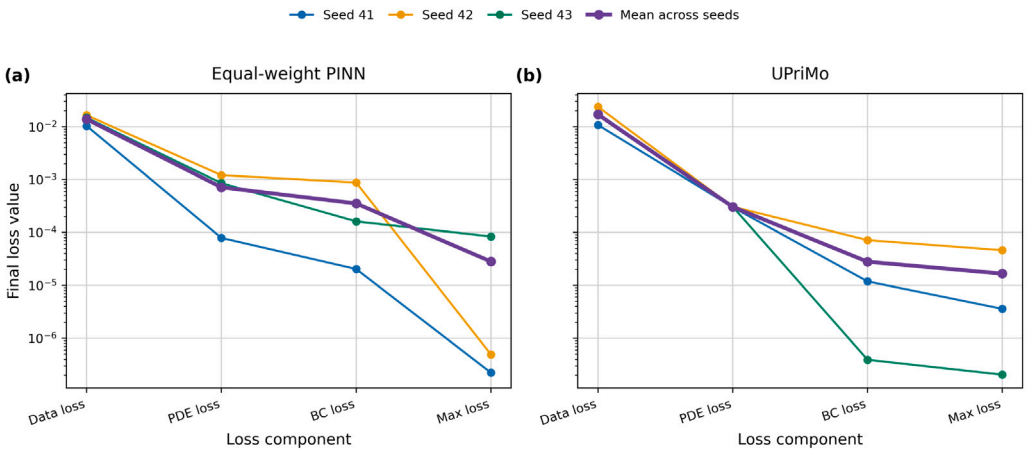


**Fig. 2.** Final loss components for the controlled laminar-flow comparison between a standard equal-weight PINN and UPriMo over three random seeds. Both models use the same architecture, noisy measurements, collocation points, and prior set; they differ only in the aggregation of the prior terms. The mean curve highlights that the utility-based objective leads to a more balanced final distribution of the loss components.

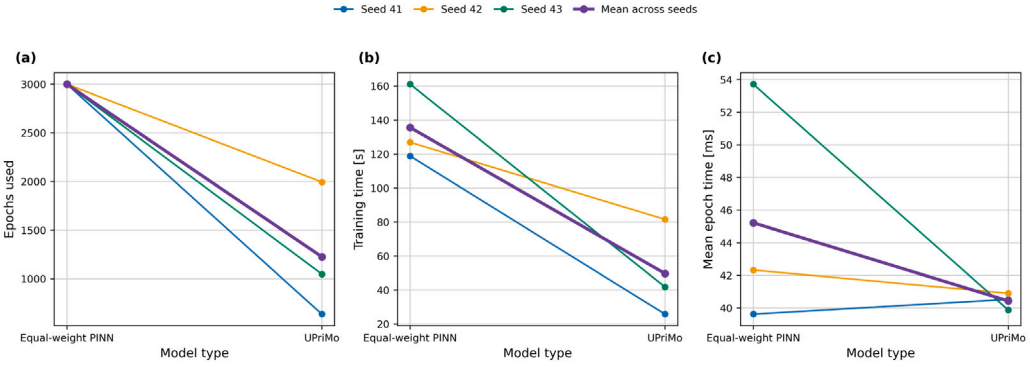


**Fig. 3.** Training-cost indicators in the controlled laminar-flow comparison: (a) number of epochs used, (b) total wall-clock training time, and (c) mean epoch time. Because both formulations evaluate the same derivative-based priors, the mean epoch cost remains of the same order, while the lower total training time of UPriMo is primarily associated with faster convergence.

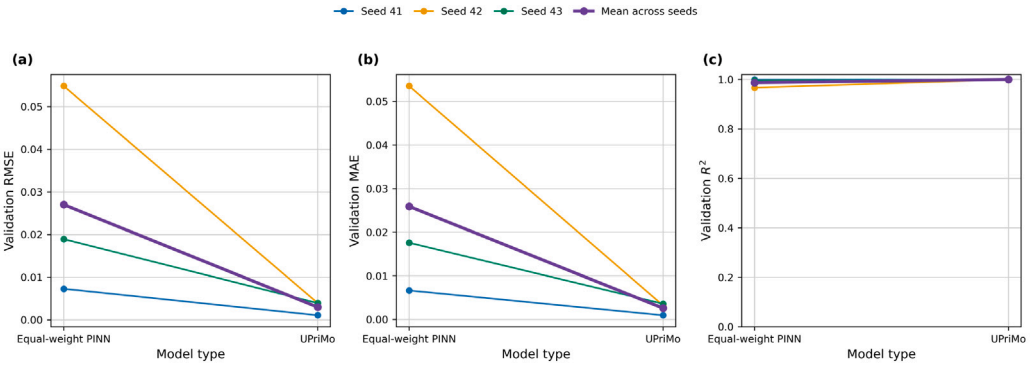


**Fig. 4.** Validation performance of the equal-weight PINN and UPriMo on the dense analytical reference grid for three random seeds: (a) RMSE, (b) MAE, and (c) $R^2$. Across all seeds, UPriMo achieves lower validation error while maintaining near-perfect agreement with the analytical profile.

A static feedforward neural network $f(\cdot;\theta)$ implements a nonlinear map over the regressor,

$$\hat{y}_k = f(\mathbf{x}_k;\theta), \tag{3.18}$$

and we adopt an output–error description

$$y_k = \hat{y}_k + e_k, \tag{3.19}$$

where $e_k$ is the identification residual minimized during one-step-ahead supervised training on the measured sequence. Free-run (autoregressive) roll-outs are evaluated separately for validation.

In the experiments we set $T_s = 2$ s, $n_a = 2$, $n_{b,u} = n_{b,q} = 2$, and equal input delays $d_u = d_q =: d$ with $d = 2$ samples. With these choices the NARX regressor specializes to

$$\mathbf{x}_k = \left[u_{k-1-d},\, u_{k-2-d},\, q_{k-1-d},\, q_{k-2-d},\, y_{k-1},\, y_{k-2}\right],$$

and the predictor remains $\hat{y}_k = f(\mathbf{x}_k;\theta)$ as in (3.18).

Within the *UPriMo* framework, prior knowledge is encoded as bounded utilities obtained from auxiliary loss terms that augment the data-fit term. We evaluate priors over two sets: (i) a *steady-state grid* $\mathcal{X}^{(c)} = \{\mathbf{x}_\ell^{(c)}\}_{\ell=1}^{N_c}$ with cardinality $|\mathcal{X}^{(c)}| = N_c = 77$, consisting of

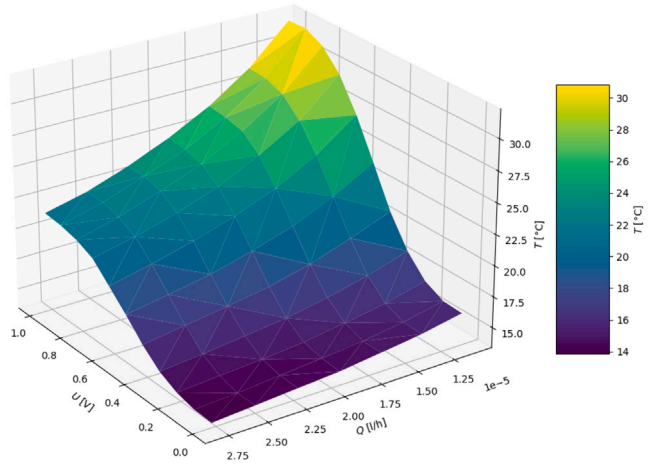


**Fig. 5.** Steady-state outlet-temperature surface $T_{\text{out}}(u, q)$ as a function of heater voltage $u$ and flow rate $q$. Grid nodes mark the $N_c = 77$ fixed operating points; the smooth surface corresponds to the step-response reduced-order reference model $f^{\text{R}}(\cdot; \theta^{\text{R}})$.

constant operating points $\mathbf{x}_\ell^{(c)} = [\, u_\ell, u_\ell, q_\ell, q_\ell, y_\ell, y_\ell \,]$; and (ii) the *measured-data set* $\mathcal{X}^{(m)} = \{\mathbf{x}_k^{(m)}\}_{k=1}^{N_m}$ with cardinality $|\mathcal{X}^{(m)}| = N_m = 283$, constructed from the dynamic sequence. Angle brackets $\langle\cdot\rangle_{\mathcal{X}^{(m)}}$ below denote empirical averages over $\mathcal{X}^{(m)}$.

Following Section 2.3, we use a fixed-parameter reference model $f^{\text{R}}(\cdot; \theta^{\text{R}})$ — a step-response reduced-order model (ROM) — that provides steady-state outlet-temperature targets at the collocation grid points. For each $\mathbf{x}_\ell^{(c)} \in \mathcal{X}^{(c)}$, we evaluate

$$\tilde{y}_\ell^{\text{SS}} := f^{\text{R}}\big(\mathbf{x}_\ell^{(c)}; \theta^{\text{R}}\big),$$

which serves as the target in the steady-state prior. All partial derivatives below are taken with respect to the components of the NARX regressor.

**(P1) Steady-state agreement (SS).** Consistency between the UPriMo predictor and the step-response ROM on $\mathcal{X}^{(c)}$ is enforced through the mean-squared error

$$L_{\text{SS}} = \frac{1}{N_c} \sum_{\ell=1}^{N_c} \Big( f(\mathbf{x}_\ell^{(c)}; \theta) - \tilde{y}_\ell^{\text{SS}} \Big)^2. \tag{3.20}$$

Fig. 5 illustrates the collocation grid and the ROM surface used as the steady-state reference.

**(P2) Voltage monotonicity (Heat).** Increasing the heater voltage must not decrease the predicted outlet temperature. This is enforced on $\mathcal{X}^{(m)}$ via mean ReLU penalties on the partial derivatives with respect to the voltage inputs:

$$L_{\text{Heat}} = \Big\langle \text{ReLU}\big(-\partial_{u_{k-1-d}} f(\mathbf{x}_k^{(m)}; \theta)\big) \Big\rangle_{\mathcal{X}^{(m)}} + \Big\langle \text{ReLU}\big(-\partial_{u_{k-2-d}} f(\mathbf{x}_k^{(m)}; \theta)\big) \Big\rangle_{\mathcal{X}^{(m)}}. \tag{3.21}$$

**(P3) Flow-rate monotonicity (Flow).** A higher flow rate must not increase the predicted outlet temperature. This constraint is applied to $\mathcal{X}^{(m)}$, mirroring the previous implementation:

$$L_{\text{Flow}} = \Big\langle \text{ReLU}\big(\partial_{q_{k-1-d}} f(\mathbf{x}_k^{(m)}; \theta)\big) \Big\rangle_{\mathcal{X}^{(m)}} + \Big\langle \text{ReLU}\big(\partial_{q_{k-2-d}} f(\mathbf{x}_k^{(m)}; \theta)\big) \Big\rangle_{\mathcal{X}^{(m)}}. \tag{3.22}$$

**(P4) Feedback stability (Stab).** Closed-loop stability is encouraged through constraints on the implicit autoregressive gains $a_1 := \partial_{y_{k-1}} f(\mathbf{x}_k^{(m)}; \theta)$ and $a_2 := \partial_{y_{k-2}} f(\mathbf{x}_k^{(m)}; \theta)$. We impose a conservative sufficient region for AR(2) stability and non-oscillatory behavior:

$$a_1 \geq 0, \tag{3.23}$$

$$a_2 \leq 0, \tag{3.24}$$

$$|a_2| < 1, \tag{3.25}$$

$$a_1 + a_2 \leq 1, \tag{3.26}$$

$$a_2 - a_1 \leq 1, \tag{3.27}$$

and penalize violations on $\mathcal{X}^{(m)}$ using the mean of five ReLU terms:

$$\begin{aligned} L_{\text{Stab}} = \frac{1}{5} \Big( & \Big\langle \text{ReLU}\big(-a_1\big) \Big\rangle_{\mathcal{X}^{(m)}} + \Big\langle \text{ReLU}\big(|a_2| - 1\big) \Big\rangle_{\mathcal{X}^{(m)}} \\ & + \Big\langle \text{ReLU}\big(a_1 + a_2 - 1\big) \Big\rangle_{\mathcal{X}^{(m)}} + \Big\langle \text{ReLU}\big(a_2 - a_1 - 1\big) \Big\rangle_{\mathcal{X}^{(m)}} \\ & + \Big\langle \text{ReLU}\big(a_2\big) \Big\rangle_{\mathcal{X}^{(m)}} \Big). \end{aligned} \tag{3.28}$$

*Remark.* Strict inequalities are enforced softly by the ReLU functions; in practice a small margin (e.g. replace 1 by $1 - \varepsilon$, $\varepsilon \in [10^{-3}, 10^{-2}]$) avoids unit-root edge cases.

The AR(2) conditions in Eqs. (3.23)–(3.27) define a conservative sufficient region rather than the full stability set. This choice intentionally favors robust free-run behavior over maximal hypothesis-space flexibility, which is appropriate in the present identification setting where small one-step errors may accumulate over long autoregressive roll-outs. Consequently, the stability prior should be interpreted as a regularizing bias toward non-oscillatory and practically stable trajectories, not as an exact characterization of all admissible AR(2) dynamics.

Having defined the prior-specific losses $L_{\text{SS}}$, $L_{\text{Heat}}$, $L_{\text{Flow}}$ and $L_{\text{Stab}}$, we map each loss to a bounded utility on $(0, 1]$ using a common rational transform:

$$\boxed{U_i(L_i) = \frac{1}{1 + \alpha\, L_i}, \qquad i \in \{\text{SS}, \text{Heat}, \text{Flow}, \text{Stab}\}.} \tag{3.29}$$

This provides a monotone bounded mapping that places heterogeneous prior losses on a common scale and allows them to be handled more uniformly within a single objective. Since the shape parameter $\alpha$ controls the sensitivity of the utility transformation, its influence is examined separately through an $\alpha$-sensitivity study.

The one-step data term is the mean-squared error under *series–parallel (open-loop) one-step-ahead* (SP–OSA) training (measured past outputs supplied to the regressor),

$$L^{\text{data}} = \frac{1}{N_m} \sum_{k=1}^{N_m} (y_k - \hat{y}_k)^2, \qquad \hat{y}_k = f(\mathbf{x}_k; \theta), \tag{3.30}$$

while free-run (autoregressive) roll-outs are used only for validation. The aggregated UPriMo objective minimized during training is

$$L = L^{\text{data}} + \big(1 - U_{\text{SS}}\big) + \big(1 - U_{\text{Flow}}\big) + \big(1 - U_{\text{Heat}}\big) + \big(1 - U_{\text{Stab}}\big), \tag{3.31}$$

where the utilities enter additively as bounded shortfalls $1 - U_i \in [0, 1]$.

Each raw signal was linearly mapped to $[0, 1]$ using extended minima and maxima,

$$\begin{aligned} & u_{\min}^{\text{ext}} = -0.40 \text{ V}, && u_{\max}^{\text{ext}} = \ \ 1.40 \text{ V}, \\ & q_{\min}^{\text{ext}} = 1.0 \times 10^{-5} \text{ m}^3\,\text{s}^{-1} \ (= 36 \text{ L}\,\text{h}^{-1}), && q_{\max}^{\text{ext}} = 2.8 \times 10^{-5} \text{ m}^3\,\text{s}^{-1} \\ & && \quad (= 101 \text{ L}\,\text{h}^{-1}), \\ & T_{\min}^{\text{ext}} = \ \ 5 \ ^\circ\text{C}, && T_{\max}^{\text{ext}} = 35 \ ^\circ\text{C}, \end{aligned}$$

and $x_{\text{norm}} = \dfrac{x - x_{\min}^{\text{ext}}}{x_{\max}^{\text{ext}} - x_{\min}^{\text{ext}}}$. By widening the bounds slightly beyond the observed extrema, most samples fall well inside a numerically well-conditioned range for the activation functions.

The control signals $(u, q)$ consist of piecewise-constant segments with occasional set-point changes within this operating envelope, reflecting typical heater operation. Both the purely data-driven NARX

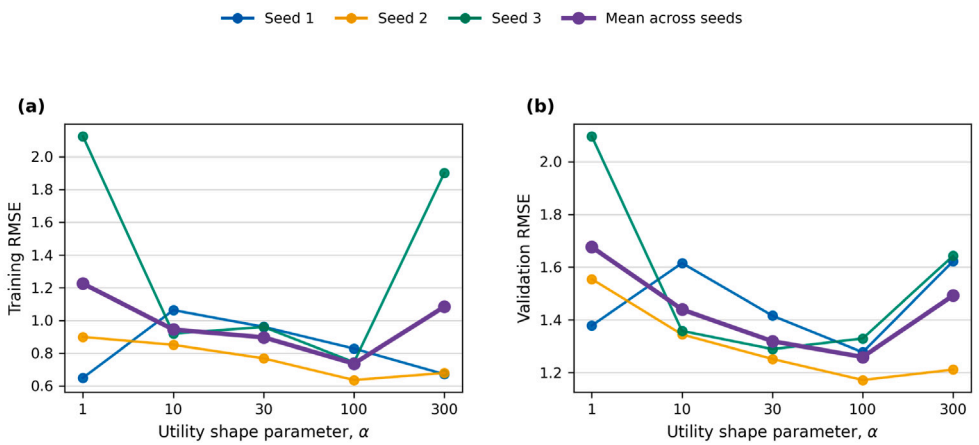


**Fig. 6.** Sensitivity of the electric-water-heater NARX model to the utility-shape parameter $\alpha$ for a representative architecture with three hidden layers and 15 neurons per layer. (a) Free-run training RMSE and (b) free-run validation RMSE on an unseen sequence, shown for three random seeds together with the mean across seeds. Validation performance improves from small to intermediate values of $\alpha$, with the most favorable mean result obtained at $\alpha = 100$, while $\alpha = 300$ leads to increased variability and degraded average performance.

(DD) and the UPriMo-regularized NARX were trained on the same nine-member architecture grid as in the laminar case (Table 3, $L \in \{1, 2, 3\}$, $H \in \{5, 10, 15\}$). Two complementary evaluation modes gauge accuracy and robustness:

1. **Closed-loop (free-run) simulation on the training sequence**, where the one-step predictor is rolled out autoregressively and prediction errors accumulate over time.
2. **Closed-loop simulation on an unseen validation sequence** with randomly generated inputs, quantifying generalization. Unless stated otherwise, the reported MSE, RMSE, MAE and $R^2$ correspond to this free-run validation setting.

After defining the aggregated UPriMo objective in Eq. (3.30), we briefly examine the sensitivity of the water-heater model to the common utility-shape parameter $\alpha$, which determines how strongly prior losses are transformed into bounded utility shortfalls. To isolate this effect, we consider a representative NARX architecture with three hidden layers and 15 neurons per layer and keep all other ingredients of the training protocol fixed, including the data split, optimizer settings, collocation grid, and prior definitions. The study is carried out for

$$\alpha \in \{1,\ 10,\ 30,\ 100,\ 300\},$$

and each configuration is repeated for three random seeds.

For each run, we evaluate free-run performance on both the training and unseen validation sequences. In addition to the training and validation RMSE, we monitor the generalization gap

$$\Delta_{\text{gen}} = \text{RMSE}_{\text{val}} - \text{RMSE}_{\text{train}},$$

as well as the aggregate prior-violation measure

$$L_{\text{prior,sum}} = L_{\text{SS}} + L_{\text{Flow}} + L_{\text{Heat}} + L_{\text{Stab}}.$$

These quantities characterize the trade-off induced by $\alpha$ between predictive accuracy and prior satisfaction.

As shown in Fig. 6, increasing $\alpha$ from very small values improves the free-run validation accuracy, with the most favorable mean validation RMSE obtained at $\alpha = 100$. At the same time, the range $\alpha \in [30, 100]$ yields comparable validation performance across seeds, indicating that the method is not excessively sensitive to the exact parameter value within this interval. For the largest tested value, $\alpha = 300$, the mean validation performance deteriorates and the seed-to-seed variability increases.

This trend is consistent with the behavior of the generalization gap and the aggregate prior violation in Fig. 7. Intermediate values of $\alpha$ maintain a favorable balance between predictive performance and prior satisfaction, whereas excessively large values lead to less robust behavior across random initializations. Based on this trade-off, $\alpha = 100$ was retained in the reported water-heater experiments.

**Table 5**
Validation results (MSE and $R^2$) for DD vs. UPriMo (free-run mode).

| Model ID | DD MSE | DD $R^2$ | UPriMo MSE | UPriMo $R^2$ |
|---|---|---|---|---|
| A | 63.396 | −2.534 | 1.786 | 0.900 |
| B | 468.701 | −25.127 | 3.172 | 0.823 |
| C | 47.724 | −1.660 | 1.903 | 0.894 |
| D | **15.550** | **0.133** | 2.703 | 0.849 |
| E | 41.192 | −1.296 | 2.016 | 0.888 |
| F | 68.606 | −2.824 | 1.927 | 0.893 |
| G | 44.946 | −1.505 | 2.015 | 0.888 |
| H | 50.196 | −1.798 | 2.247 | 0.875 |
| I | 49.647 | −1.767 | **1.690** | **0.906** |

We evaluate nine architectures for two model types:

1. *purely data-driven NARX* (DD), trained only on the data term $L^{\text{data}}$;
2. *UPriMo-regularized NARX* (UPriMo), trained using the composite loss (3.31).

Table 5 summarizes validation MSE and $R^2$ across the architecture grid (free-run mode on the unseen sequence). The best DD model (D: 2 layers, 5 neurons) attains MSE = 15.55, $R^2 = 0.133$, whereas the best UPriMo model (I: 3 layers, 15 neurons) achieves MSE = 1.690, $R^2 = 0.906$. Moreover, UPriMo improves every architecture relative to its DD counterpart, indicating a consistent advantage.

Figs. 8 and 9 show validation free-run trajectories (overlay of DD, UPriMo and ground truth) on a horizon of ≈2800 s (about 1400 samples, $t = kT_s$, $T_s = 2$ s). The DD model exhibits gradual drift and error accumulation after large set-point changes and during prolonged plateaus. In contrast, UPriMo tracks step transients and steady segments closely, without runaway bias. Crucially, the UPriMo trajectories remain physically plausible — no *non-monotone* response to heater voltage and no *spurious heating* under increased flow — consistent with the (P2)–(P3) priors, and display well-damped autoregressive behavior consistent with (P4).

The utility-balanced priors regularize the predictor in directions that are physically meaningful. The steady-state prior (P1) anchors the NARX map on the measured operating envelope, reducing bias on long plateaus. Voltage monotonicity (P2) and flow anti-monotonicity (P3) shape the local gradients $\partial f/\partial u$ and $\partial f/\partial q$, eliminating nonphysical

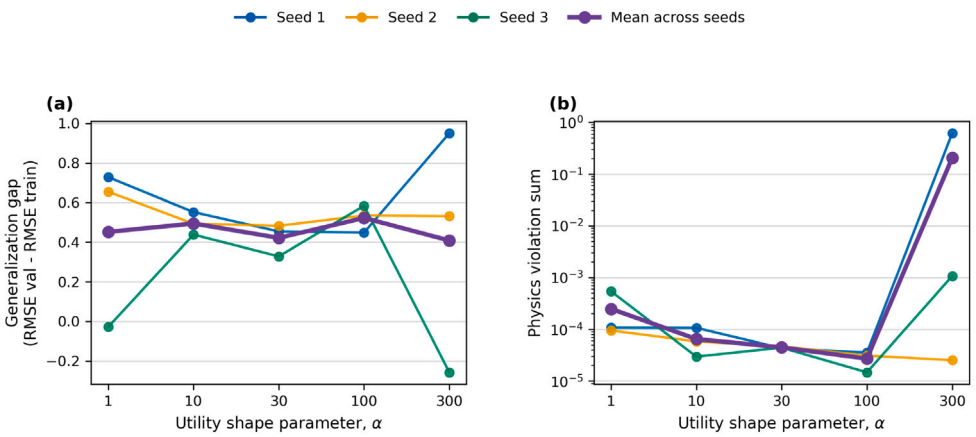


**Fig. 7.** Additional summary of the $\alpha$-sensitivity study for the same representative electric-water-heater NARX model. (a) Generalization gap, defined as $\mathrm{RMSE}_{\mathrm{val}} - \mathrm{RMSE}_{\mathrm{train}}$, and (b) aggregate prior violation $L_{\mathrm{SS}} + L_{\mathrm{Flow}} + L_{\mathrm{Heat}} + L_{\mathrm{Stab}}$, shown for three random seeds together with the mean across seeds. Intermediate values of $\alpha$ provide a favorable compromise between predictive performance and prior satisfaction, whereas the largest tested value yields less robust behavior.

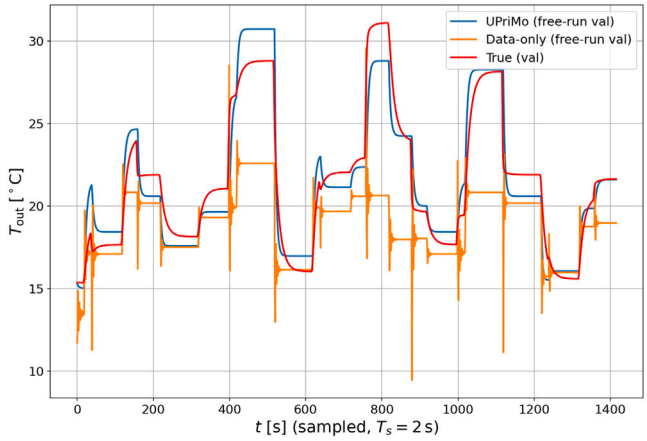


**Fig. 8.** Validation free-run comparison (Model D: 2×5). Horizontal axis: time $t$ in seconds, sampled at $T_s = 2$ s (equivalently $k = t/T_s$). Vertical axis: outlet temperature $T_{\mathrm{out}}$ [°C]. The purely data-driven model (DD) accumulates error over long horizons, whereas UPriMo remains stable and accurate.

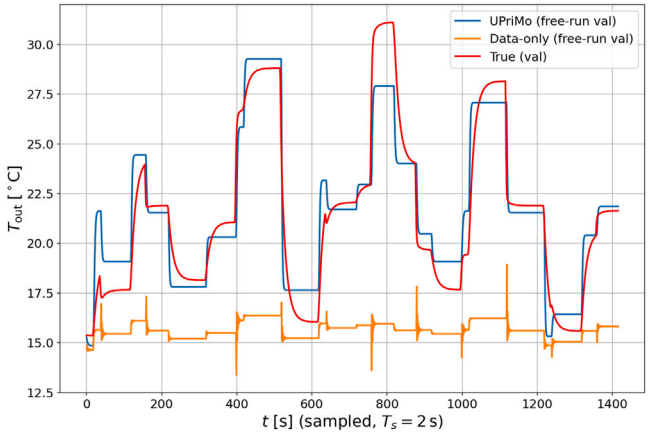


**Fig. 9.** Validation free-run comparison (Model I: 3×15). Same axes as in Fig. 8. UPriMo preserves long-horizon accuracy and honors physical constraints (monotonicity, feedback stability), preventing the drift seen in the DD model.

inversions that trigger compounding free-run errors. The AR(2) stability prior (P4) restricts the implicit feedback gains $(a_1, a_2)$ in autoregressive roll-outs, suppressing oscillatory growth and drift. These effects are directly visible in Figs. 8–9: UPriMo responds sharply yet without overshoot when $u$ steps up, cools appropriately when $q$ increases and maintains bounded dynamics over ∼1400 steps.

Beyond average metrics, priors substantially reduce variance across architectures: several DD configurations (e.g., B, H) degrade catastrophically (negative $R^2$), while the corresponding UPriMo models remain accurate and stable. This indicates that the priors act as *guardrails*, converting ill-posed regions of the parameter space into feasible ones by penalizing violations in a scale-free manner via the utilities $U_i = (1 + \alpha L_i)^{-1}$.

UPriMo does not replace data: performance still depends on excitation richness (e.g., sufficient variation in operating set-points) and on the relevance of priors. Also, priors encode sufficient conditions (e.g., a conservative AR(2) stability region) and may constrain the hypothesis space more than necessary. Nevertheless, the results demonstrate that blending lightweight physics and expert heuristics with data enables the model to internalize the system's qualitative behavior and prevents long-horizon error accumulation.

## 4. Conclusion and outlook

We introduced the *Utility-balanced Prior-informed Model* (UPriMo), a unified objective that turns heterogeneous prior knowledge into bounded utilities and combines them additively with the data misfit. By mapping each prior loss $L_i$ to a bounded utility $U_i = (1+\alpha L_i)^{-1}$, UPriMo places heterogeneous prior terms on a common scale, keeps their contribution bounded during training, and can be wrapped around any differentiable predictor. The framework accommodates steady-state consistency relations, structural and topological information, monotonic trends, stability surrogates and expert heuristics within a single training recipe.

We demonstrated the approach on two representative energy problems. In the stationary laminar pipe-flow case, UPriMo enabled accurate reconstruction of the velocity profile from extremely sparse and noisy measurements, while preserving physical features such as symmetry, zero wall velocity and a single centerline maximum. In the electric water-heater case, a discrete-time NARX surrogate was regularized by steady-state, monotonicity and stability priors. Across architectures, UPriMo consistently improved long-horizon free-run behavior compared with purely data-driven baselines, suppressing drift, nonphysical responses and unstable roll-outs. In both studies, the priors acted as guardrails that helped the surrogate internalize the system's

qualitative behavior and maintain physically plausible responses within the operating envelope.

From a practical standpoint, UPriMo integrates seamlessly with standard machine-learning workflows: it uses the same optimizers and batching schemes as the baseline learner, adds only modest overhead for gradient-based prior terms and is compatible with common preprocessing and normalization strategies. The bounded utility terms ensure that no single prior can dominate training, yet violations are consistently penalized. This makes UPriMo particularly attractive for energy applications where data are scarce, physics are partially known and trustworthy long-horizon simulations are essential for predictive control, digital twins and asset monitoring.

In the water-heater case study, the main benefit of UPriMo is not only improved one-step accuracy, but also stable and physically plausible long-horizon free-run behavior. This is particularly relevant for model-based control, where the learned surrogate serves as the prediction model in MPC and repeated multi-step roll-outs are more critical than one-step fit alone. A similar requirement appears in state estimation, where the model enters the prediction step of observer-based schemes. In this sense, the present paper addresses the quality of the modeling layer needed for downstream control and estimation, while closed-loop demonstrations remain a natural direction for future work.

The present validation focuses on long-horizon free-run stability and physical plausibility within the operating envelopes represented in the training and collocation data. Robustness under stronger distribution shift and explicit predictive uncertainty quantification were not addressed here and remain important directions for deployment-oriented extensions, especially in digital-twin and decision-support settings. Natural next steps include uncertainty-aware variants of UPriMo, for example through ensemble-based, Bayesian, or conformal wrappers, and testing under deliberately shifted operating regimes.

If the prior knowledge is conflicting or incorrectly specified, the method does not resolve this automatically, and the learned model may be biased by such assumptions. The bounded utility formulation limits the influence of each prior term, but it does not make unsuitable priors harmless or guarantee that all priors can be satisfied simultaneously. In more complex systems, the design of informative priors may also become less transparent, and selecting useful priors may require more domain knowledge and iteration. The practical success of the framework therefore still depends strongly on the relevance and consistency of the selected priors.

An important next step is the extension of the framework to more complex systems. One possible direction is to decompose such systems into interconnected cascade-like subsystems, where the output of one block serves as the input of the next. In this modular setting, prior knowledge such as PDE-based relations, physical laws, or parameter constraints could be incorporated at the level of the individual submodels. This may provide a practical route toward applying the framework to larger and more structured process systems.

Further directions include learning or adapting the utility shape parameter $\alpha$ online, extending the library of priors to structural and graph constraints in multi-unit energy systems, and hybridizing with physics solvers for tighter guarantees. Overall, the evidence across both benchmarks suggests that encoding lightweight physics and expert knowledge through utilities is a promising path toward data-efficient, reliable AI.

## CRediT authorship contribution statement

**András Edvy:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology. **Alex Kummer:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **János Abonyi:** Writing – review & editing, Supervision, Resources, Methodology, Investigation, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors 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

This project was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences (B0/00439/25/6).

## Data availability

Data will be made available on request.

## References

[1] A. Brown, A. Foley, D. Laverty, S. McLoone, P. Keatley, Heating and cooling networks: A comprehensive review of modelling approaches to map future directions, Energy 261 (Part B) (2022) 125060, http://dx.doi.org/10.1016/j.energy.2022.125060.

[2] S. Kuntuarova, T. Lickleder, T. Huynh, D. Zinsmeister, T. Hamacher, V. Perić, Design and simulation of district heating networks: A review of modeling approaches and tools, Energy 305 (2024) 132189, http://dx.doi.org/10.1016/j.energy.2024.132189.

[3] G. Schweiger, R. Heimrath, B. Falay, K. O'Donovan, P. Nageler, R. Pertschy, G. Engel, W. Streicher, I. Leusbrock, District energy systems: Modelling paradigms and general-purpose tools, Energy 164 (2018) 1326–1340, http://dx.doi.org/10.1016/j.energy.2018.08.193.

[4] J. Zou, T. Hirokawa, J. An, L. Huang, J. Camm, Recent advances in the applications of machine learning methods for heat exchanger modeling—A review, Front. Energy Res. 11 (2023) 1294531, http://dx.doi.org/10.3389/fenrg.2023.1294531.

[5] S. Du, L. Jin, Z. Huang, X. Wan, Recent progress in hybrid intelligent modeling technology and optimization strategy for industrial energy consumption processes, Energies 18 (8) (2025) 1939, http://dx.doi.org/10.3390/en18081939.

[6] C.L. Gargalo, A.A. Malanca, A.R.N. Aouichaoui, J.K. Huusom, K.V. Gernaey, Navigating industry 4.0 and 5.0: The role of hybrid modelling in (bio)chemical engineering's digital transition, Front. Chem. Eng. 6 (2024) 1494244, http://dx.doi.org/10.3389/fceng.2024.1494244.

[7] D. Jalili, S. Jang, M. Jadidi, G. Giustini, A. Keshmiri, Y. Mahmoudi, Physics-informed neural networks for heat transfer prediction in two-phase flows, Int. J. Heat Mass Transfer 221 (2024) 125089, http://dx.doi.org/10.1016/j.ijheatmasstransfer.2023.125089.

[8] Z. Wu, M. Li, C. He, B. Zhang, J. Ren, H. Yu, Q. Chen, Physics-informed learning of chemical reactor systems using decoupling–coupling training framework, AIChE J. 70 (7) (2024) e18436, http://dx.doi.org/10.1002/aic.18436.

[9] L. Boca de Giuli, A. La Bella, R. Scattolini, Physics-informed neural network modelling and predictive control of district heating systems, 2023, http://dx.doi.org/10.48550/arXiv.2310.13937, arXiv Preprint arXiv:2310.13937.

[10] B. Sohlberg, Grey box modelling for model predictive control of a heating process, J. Process Control 13 (3) (2003) 225–238, http://dx.doi.org/10.1016/S0959-1524(02)00030-6.

[11] B. Zhu, S. Ren, Q. Weng, F. Si, A physics-informed neural network that considers monotonic relationships for predicting $NO_x$ emissions from coal-fired boilers, Fuel 364 (2024) 131026, http://dx.doi.org/10.1016/j.fuel.2024.131026.

[12] P. Li, F. Guo, Y. Li, X. Yang, X. Yang, Physics-informed neural network for real-time thermal modeling of large-scale borehole thermal energy storage systems, Energy 315 (2025) 134344, http://dx.doi.org/10.1016/j.energy.2024.134344.

[13] V.R. Chifu, T. Cioara, C.B. Pop, I. Anghel, A. Pelle, Physics-informed neural networks for heat pump load prediction, Energies 18 (1) (2025) 8, http://dx.doi.org/10.3390/en18010008.

[14] M. Raissi, P. Perdikaris, G.E. 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.

[15] S. Cai, Z. Mao, Z. Wang, M. Yin, G.E. Karniadakis, Physics-informed neural networks (PINNs) for fluid mechanics: A review, Acta Mech. Sin. 37 (12) (2021) 1729–1740, http://dx.doi.org/10.1007/s10409-021-01148-1.

[16] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics–Informed neural networks: Where we are and what's next, J. Sci. Comput. 92 (2022) http://dx.doi.org/10.1007/s10915-022-01939-z.

[17] J. Abonyi, Fuzzy Model Identification for Control, Springer Science & Business Media, 2003.