Physics-Informed Non-Dimensionalisation for Neural Solvers

Jochen Stiasny, Jochen L. Cremer

Delft University of Technology, AIT Austrian Institute of Technology

Suggested Citation

J. Stiasny and J. L. Cremer, "Physics-Informed Non-Dimensionalisation for Neural Solvers," in *Proceedings of the PowerUp Conference*, 2026.

BibTeX Entry

```
@inproceedings{stiasny2026physicsinformed,
  author = {Stiasny, Jochen and Cremer, Jochen L.},
  title = {Physics-Informed Non-Dimensionalisation for Neural Solvers},
  booktitle = {Proceedings of the PowerUp Conference},
  year = {2026},
}
```

Physics-Informed Non-Dimensionalisation for Neural Solvers

Jochen Stiasny, Jochen L. Cremer

***Abstract*—The ability to assess and control approximation errors is key to the success of numerical methods. In contrast, learned approximations must ensure during training that error requirements are met, typically relying on statistical loss functions that do not account for the underlying physics. In this work, we argue that the assessment of learning errors in neural solvers should be aligned with numerical methods by evaluating errors in the residual norm of the governing equations. We show how this residual-based error measure can be integrated into the learning process through a physics-informed non-dimensionalisation of inputs and outputs, yielding a training objective that reflects physical sensitivities. The proposed construction reduces reliance on dataset-dependent statistics, improves robustness for limited training data, and provides an interpretable connection between training loss and engineering error measures. We demonstrate the approach on learning power flow solutions as an example of neural solvers for algebraic systems.**

***Index Terms*—Neural solver, numerical methods, error estimation, power flow**

I. Introduction

The success of numerical methods in engineering is rooted in their ability to systematically assess and control approximation errors. Classical solvers are designed around residual norms, convergence guarantees, and tolerance criteria that allow practitioners to trade computational cost for accuracy in a principled way [1]–[3]. In contrast, learned approximations of numerical solutions—we use the umbrella-term *neural solvers*—have been proposed to directly approximate the solution operator of a physical system [4]–[7]. Their performance is predominantly determined by empirical risk minimisation during training [8]. These loss functions, typically mean-squared or absolute errors, often treat all prediction components as commensurate quantities that can be directly compared. While effective for optimisation, such losses abstract away physical units and sensitivities inherent to the underlying system. As a result, the training objective implicitly defines a trade-off between heterogeneous errors, yet this trade-off is rarely aligned with error notions traditionally used in numerical analysis or engineering practice.

Let us consider a concrete example from power flow analysis: at a certain bus, an approximation exhibits a voltage magnitude error of $0.001\,\mathrm{p.u.}$, a voltage angle error of 0.001° and an active power imbalance of $0.001\,\mathrm{p.u.}$. Different physical units and sensitivities make the error assessment non-trivial, in fact they lead to a multi-objective optimisation problem. The user of a numerical method can specify element-wise tolerances, and if these are not met, additional iterations or refinements are performed. In contrast, the user of a neural solver is fundamentally limited by the accuracy achieved during the training phase.

The solution of multi-objective optimisation problems is a well-studied field and often framed around the discovery of Pareto-optimal solutions [9]–[11]. Learning problems in which multiple targets are simultaneously approximated, such as for physical systems, naturally align with this framing [12]. Varied gradient updates [13] and adaptive loss weighting [14] have been proposed to handle the multi-objective character. For a few objectives, this works well, but a large number of objectives or tasks become challenging. For physical systems such as power systems, prediction tasks can easily involve hundreds of target variables. Hence, *scalarisation* is typical. The multiple objectives are reduced into a single scalar by weighting the separate objectives. Accompanied by the practice of standardisation or normalisation of the input and output variables [15], learning problems can be effectively solved. However, the weighting and dataset statistics implicitly influence the learning objective. When learning on a physical system, such practices neglect the governing physical relations.

The approach of physics-informed neural networks (PINNs) incorporates the governing equations in a loss function [16]–[18]. While this loss penalises non-physical learned solutions, the balancing of the loss terms and gradients still poses challenges to the training [19]–[21]. The origin and most prominent part of research on PINNs stems from the field of fluid dynamics [17]. The field applies the modelling strategy of non-dimensionalisation [22], [23]. This approach scales variables according to characteristic units and removes physical units in order to make variables better comparable and identify similarity between phenomena. This non-dimensionalisation enabled a common error assessment in the fluid-related PINN literature, and its importance for learning is being investigated [24]. In power systems, such rigorous structure is less common and hence the performance assessment varies significantly, e.g., among neural solvers for power flow [18], [25], [26] or optimal power flow [27]–[32]. The lack of a common assessment strategy leads back to the multi-objective nature of neural solvers for physical systems: how should prediction errors across different physical variables be compared or prioritised. For optimal power flow, additional objectives and constraints further complicate this comparison. A formal verification procedure [33] can help to bound approximation errors [34], but since it is performed after training, the multi-objective structure during training remains.

In this work, we propose a physics-informed approach to comparing prediction errors in neural solvers for algebraic systems, grounded in the principles of numerical analysis. Rather than scalarising prediction errors based on dataset statistics or training dynamics, we derive a norm for error assessment directly from the residual formulation of the underlying physical equations. By aligning the learning objective

The authors are with the Department of Electrical Sustainable Energy, Delft University of Technology, Delft, Netherlands and the Austrian Institute of Technology AIT GmbH, Vienna, Austria. Email: j.b.stiasny@tudelft.nl, j.l.cremer@tudelft.nl

with the residual-based objectives used in numerical schemes, we obtain a non-dimensionalisation of inputs and outputs that induces a physically meaningful scalar training objective. In contrast to PINNs, the governing equations thus enter through a non-dimensionalisation of the variables rather than through an additional loss term. This construction reduces reliance on dataset statistics, improves robustness for limited training data, and provides an interpretable connection between training loss and engineering error measures. We demonstrate the approach on learning power flow solutions, where multiple heterogeneous variables must be predicted consistently.

The remainder of the paper is organised as follows. Section II presents the methodological framework and derives the proposed non-dimensionalisation strategy. Section III describes the case study and experimental setup. Section IV reports and discusses the results. Section V concludes.

II. Methodology

The objective of this section is to derive a physics-informed non-dimensionalisation for neural solvers that aligns their error assessment with the assessment used in numerical methods. We construct affine transformations of inputs and outputs such that a standard unweighted mean-squared error in the transformed variables corresponds to a physically meaningful error measure in the original variables. To this end, we compare the learning problem and the numerical solution problem at the level of their local quadratic structure, which allows us to derive output scaling from residual sensitivities and input scaling from backward error considerations.

A. Solution approximation of an algebraic physical system

We consider a physical system defined by a set of algebraic equations $\boldsymbol{F} \in \mathbb{R}^q$

$$\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}. \tag{1}$$

We distinguish between inputs $\boldsymbol{x} \in \mathbb{R}^m$, which parametrise the physical system and are not optimised by the solvers, and outputs $\boldsymbol{y} \in \mathbb{R}^n$, which are the variables solved for numerically and approximated by the neural solver. The corresponding *solution* mapping, which we assume to be injective on the relevant domain, is denoted as

$$\boldsymbol{\Phi} : \boldsymbol{x} \mapsto \boldsymbol{y} \tag{2}$$

and the goal of a neural solver $\hat{\boldsymbol{\Phi}}_\theta$ or numerical solver $\hat{\boldsymbol{\Phi}}_*$ is to approximate the outputs $\hat{\boldsymbol{y}}$

$$\hat{\boldsymbol{\Phi}}_\theta, \hat{\boldsymbol{\Phi}}_* : \ \boldsymbol{x} \mapsto \hat{\boldsymbol{y}}. \tag{3}$$

While the approximation target is identical, the construction of the respective solver differs.

Regarding the assessment of the approximation quality, we compare element-wise as this yields an interpretable quantity that aligns with engineering practices. As a result, each output variable y_i has a performance metric $\mathcal{P}_i$, most commonly the maximum approximation error across the input domain of interest $\mathcal{X}$

$$\mathcal{P}_i(\boldsymbol{x}) := \left\| y_i(\boldsymbol{x}) - \hat{y}_i(\boldsymbol{x}) \right\|_\infty, \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{4}$$

The maximum error yields an evaluation that is most relevant for hard constraints and limits. For risk-based assessments, a quantile analysis or other norms could be employed. Searching for "good" approximations requires consolidating the performance assessments for the different output variables y_i

$$\min_{\hat{\boldsymbol{\Phi}}} \ \{\mathcal{P}_1(\boldsymbol{x}); \mathcal{P}_2(\boldsymbol{x}); \ldots; \mathcal{P}_n(\boldsymbol{x})\} \tag{5}$$

which therefore poses a multi-objective optimisation problem.

B. Solution approximation as a learning problem

We formulate a standard supervised learning problem to learn the neural solver $\hat{\boldsymbol{\Phi}}_\theta$. The neural solver is an instance of a hypothesis class that is parametrised by the parameters θ

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{\Phi}}_\theta(\boldsymbol{x}). \tag{6}$$

The hypothesis class defines the architecture, for example, a feed-forward neural network, a graph neural network, or an operator network. Based on a training dataset

$$\mathcal{D}_{\text{train}} = \left\{ \left(\boldsymbol{x}^{(j)}; \boldsymbol{y}^{(j)} \right) \right\}_{1 \le j \le |\mathcal{D}_{\text{train}}|} \tag{7}$$

and the loss function

$$\mathcal{L}(\mathcal{D}) = \frac{1}{2} \frac{1}{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{D}|} \left(\hat{\boldsymbol{y}}^{(j)} - \boldsymbol{y}^{(j)} \right)^\top W_{\boldsymbol{y}} \left(\hat{\boldsymbol{y}}^{(j)} - \boldsymbol{y}^{(j)} \right) \tag{8}$$

with the weighting matrix $W_{\boldsymbol{y}} \in \mathbb{R}^{n \times n}$, we optimise

$$\min_\theta \quad \mathcal{L}(\mathcal{D}_{\text{train}}; \theta) \tag{9a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{y}} = \hat{\boldsymbol{\Phi}}_\theta(\boldsymbol{x}). \tag{9b}$$

As (9) can overfit to the training dataset, we actually aim to optimise the performance on the validation dataset

$$\min_{\hat{\boldsymbol{\Phi}}_\theta, W_{\boldsymbol{y}}} \quad \mathcal{L}(\mathcal{D}_{\text{validation}}; \theta^*) \tag{10a}$$

$$\text{s.t.} \quad \theta^* = \arg\min_\theta \mathcal{L}(\mathcal{D}_{\text{train}}) \tag{10b}$$

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{\Phi}}_\theta(\boldsymbol{x}). \tag{10c}$$

The hyperparameter optimisation problem (10) is usually approximated based on a few hyperparameter samples due the computational effort and non-differentiability of the training process. While the hypothesis class for $\hat{\boldsymbol{\Phi}}_\theta$ determines the approximation capacity and is regularly tuned, the effect of $W_{\boldsymbol{y}}$ is rarely considered. However, the selection of $W_{\boldsymbol{y}}$ defines the scalarisation of the multi-objective problem and therefore, has immediate consequences on the implied error trade-offs.

C. Solution approximation as a numerical problem

Numerical schemes begin by formulating the algebraic constraints as a set of residual functions $\boldsymbol{r} \in \mathbb{R}^q$

$$\boldsymbol{r} := \boldsymbol{F}(\boldsymbol{x}, \hat{\boldsymbol{y}}). \tag{11}$$

We then define the residual norm ρ

$$\rho := \frac{1}{2} \boldsymbol{r}^\top W_{\boldsymbol{r}} \boldsymbol{r} \tag{12}$$

with a weighting matrix $W_{\boldsymbol{r}}$ of the residuals. We set $W_{\boldsymbol{r}} = I$ throughout, so that all residual components enter uniformly; we note, that this itself is a modelling choice that could be used for problem-specific adaptations. A generic numerical scheme aims to solve

$$\min_{\hat{\boldsymbol{y}}} \quad \rho, \tag{13}$$

most commonly through iterative updates. A Taylor-series expansion of ρ around the current value $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ yields

$$\begin{aligned}\rho(\boldsymbol{x}, \boldsymbol{y}) \approx \rho(\boldsymbol{x}_0, \boldsymbol{y}_0) + \left(\frac{\partial \rho}{\partial(\boldsymbol{x}, \boldsymbol{y})}\bigg|_{\boldsymbol{x}_0, \boldsymbol{y}_0} \right)^\top \begin{bmatrix}\Delta\boldsymbol{x} \\ \Delta\boldsymbol{y}\end{bmatrix} \\ + \frac{1}{2} \begin{bmatrix}\Delta\boldsymbol{x} & \Delta\boldsymbol{y}\end{bmatrix} H \begin{bmatrix}\Delta\boldsymbol{x} \\ \Delta\boldsymbol{y}\end{bmatrix} + \mathcal{O}\left((\Delta\boldsymbol{x}, \Delta\boldsymbol{y})^3\right).\end{aligned} \tag{14}$$

with the Hessian H

$$H := \begin{bmatrix}H_{xx} & H_{xy} \\ H_{yx} & H_{yy}\end{bmatrix} = \frac{\partial^2 \rho}{\partial(\boldsymbol{x}, \boldsymbol{y})^2}\bigg|_{\boldsymbol{x}_0, \boldsymbol{y}_0} \tag{15}$$

Assuming a linear residual model, we obtain the Newton-Raphson scheme to solve (13). For physical systems, we assume that (13) can reach $\rho = 0$ or up to machine precision in digital computations. In practice, the minimisation problem is terminated when a tolerance setting $\varepsilon_r \in \mathbb{R}$ is met

$$\|\boldsymbol{r}\|_\infty < \varepsilon_r \tag{16}$$

where $\|\boldsymbol{r}\|_\infty = \max_i |r_i|$. The tolerance ε_r can be adjusted and evaluated element-wise to account for scale differences. However, as soon as the optimisation is converging, the tolerance setting mainly determines the required number of iterations and can be adjusted easily.

D. Aligning the learning and numerical problem formulation

We now derive a physics-informed non-dimensionalisation by aligning the learning objective with the residual-based optimisation problem solved by numerical schemes. The key idea is to compare the local quadratic structure of the learning loss $\mathcal{L}$ and the residual norm ρ, as this structure governs the interaction between different variables during optimisation.

In supervised learning, the weighted mean-squared error in (9) induces a quadratic form in the output prediction error $\hat{\boldsymbol{y}}^{(j)} - \boldsymbol{y}^{(j)}$,

$$\begin{bmatrix}\boldsymbol{x}^{(j)} & \hat{\boldsymbol{y}}^{(j)} - \boldsymbol{y}^{(j)}\end{bmatrix} \begin{bmatrix}0 & 0 \\ 0 & W_{\boldsymbol{y}}\end{bmatrix} \begin{bmatrix}\boldsymbol{x}^{(j)} \\ \hat{\boldsymbol{y}}^{(j)} - \boldsymbol{y}^{(j)}\end{bmatrix} \tag{17}$$

while the inputs $\boldsymbol{x}^{(j)}$ do not directly affect the loss. In contrast, numerical methods minimise the residual norm (12) whose local behaviour around a point $(\boldsymbol{x}_0, \boldsymbol{y}_0)$ is characterised by the second-order Taylor expansion

$$\begin{bmatrix}\Delta\boldsymbol{x} & \Delta\boldsymbol{y}\end{bmatrix} \begin{bmatrix}H_{xx} & H_{xy} \\ H_{yx} & H_{yy}\end{bmatrix} \begin{bmatrix}\Delta\boldsymbol{x} \\ \Delta\boldsymbol{y}\end{bmatrix} \tag{18}$$

up to higher-order terms.

Since numerical solvers optimise the outputs $\boldsymbol{y}$ while keeping the inputs $\boldsymbol{x}$ fixed, the relevant curvature governing convergence is the curvature of the residual norm with respect to the outputs. Matching the quadratic forms in (17) and (18) therefore yields

$$W_{\boldsymbol{y}} = H_{yy} \tag{19}$$

which incorporates the physical sensitivities of the residuals to perturbations in the outputs. This choice ensures that equal errors in the weighted loss correspond to equal local increases in the residual norm.

While the learning loss does not explicitly penalise input variations, the residual norm generally couples inputs and outputs. To derive a consistent transformation for the inputs, we consider the backward error interpretation of numerical analysis [1]: given an output perturbation, how much would the inputs need to change for the perturbed solution to satisfy the governing equations? This effect is captured by the Schur complement of the Hessian of the residual norm,

$$S := H_{yy} - H_{yx} H_{xx}^{-1} H_{xy} \tag{20}$$

which isolates the effective curvature with respect to the outputs when the inputs are held fixed. Instead of choosing the input transformation $W_{\boldsymbol{x}}$ simply as H_{xx}, we set

$$W_{\boldsymbol{x}} := H_{xy} H_{yy}^{-1} H_{yx} \tag{21}$$

such that the Schur complement becomes 0. Thereby, linear correlations between input variations and output prediction errors are removed, yielding a non-dimensionalisation consistent with linear backward error considerations.

Due to the nonlinearity of the physical system, the resulting transformation matrices are sample-dependent, which we indicate by the superscript (j). In practice, we compute the output and input transformations ${W_{\boldsymbol{y}}}^{(j)}$ and ${W_{\boldsymbol{x}}}^{(j)}$ for each training sample and obtain global transformations by averaging over the training dataset

$$W_{\boldsymbol{y}} = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{j=1}^{|\mathcal{D}_{\text{train}}|} {W_{\boldsymbol{y}}}^{(j)}, \; W_{\boldsymbol{x}} = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{j=1}^{|\mathcal{D}_{\text{train}}|} {W_{\boldsymbol{x}}}^{(j)} \tag{22}$$

This approximation avoids additional computational effort at inference time while preserving the dominant physical sensitivities encoded in the governing equations.

E. Physics-informed non-dimensionalisation

The alignment of the learning and numerical objectives derived in the previous subsection can be interpreted as a physics-informed non-dimensionalisation of the learning problem. Rather than introducing explicit weights in the loss function, we construct affine transformations of inputs and outputs such that a standard unweighted mean-squared error in the transformed variables corresponds to a physically meaningful error measure in the original variables.

Concretely, we introduce affine transformations

$$T_x: \quad \boldsymbol{x} = A_x \tilde{\boldsymbol{x}} + \boldsymbol{b}_x \tag{23a}$$

$$T_y: \quad \boldsymbol{y} = A_y \tilde{\boldsymbol{y}} + \boldsymbol{b}_y \tag{23b}$$

where $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$ denote non-dimensional inputs and outputs. The bias terms $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$ are set to the corresponding empirical mean of the training data, while the linear transformation matrices A_x, A_y are chosen such that

$$A_x^\top A_x = W_{\boldsymbol{x}}^{-1} \tag{24a}$$
$$A_y^\top A_y = W_{\boldsymbol{y}}^{-1}. \tag{24b}$$

Training a neural solver on the transformed variables using an unweighted mean-squared error is therefore equivalent to minimising a weighted loss in the original variables that reflects the physical sensitivities encoded in the governing equations. In this sense, the proposed approach shifts the design choice from selecting loss weights to defining a physically motivated non-dimensionalisation.

More generally, the affine transformations define a non-dimensionalisation for any choice of A_x and A_y. The proposed physics-informed scheme is the special case in which they are set from the residual sensitivities $W_{\boldsymbol{x}}$ and $W_{\boldsymbol{y}}$ derived above; the identity and standardisation strategies correspond to other choices of A_x and A_y, which we detail in Section III-A.

III. Case Study

The case study evaluates the proposed physics-informed non-dimensionalisation for learning power flow solutions and is designed to assess three aspects: (i) the alignment between the training objective and the residual norm used in numerical methods, (ii) robustness with respect to limited training data, and (iii) sensitivity to the distribution of the training dataset.

We focus here on the key aspects of the study design and describe the full setup in the Appendix.

A. Non-dimensionalisation strategies

We test three choices for the transformations T_x and T_y: identity, standardised, and physics-informed. All centre the variables on the training mean, $\boldsymbol{b}_x = \mathbb{E}[\boldsymbol{x}]$ and $\boldsymbol{b}_y = \mathbb{E}[\boldsymbol{y}]$, and differ only in the scaling matrices A_x and A_y summarised in Table I.

Scheme	A_x	A_y
Identity	I	I
Standardised	$\mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{x}})^{-1}$	$\mathrm{diag}(\boldsymbol{\sigma}_{\boldsymbol{y}})^{-1}$
Physics-informed	$W_{\boldsymbol{x}}^{-1/2}$	$W_{\boldsymbol{y}}^{-1/2}$

TABLE I: Non-dimensionalisation schemes. Standardisation scales by the dataset's element-wise standard deviations $\boldsymbol{\sigma}_{\boldsymbol{x}}, \boldsymbol{\sigma}_{\boldsymbol{y}}$; the physics-informed scheme uses the residual-sensitivity matrices $W_{\boldsymbol{x}}, W_{\boldsymbol{y}}$.

Identity treats all variables as equal by dropping their physical units. Standardisation removes scale differences, but is sensitivity-blind and strongly dataset-dependent. The physics-informed scheme instead derives the scaling from the residual sensitivities; it is the only scheme whose A_x, A_y carry off-diagonal terms, capturing the correlations between variables. It thereby aligns the training objective with the residual norm used in numerical methods and depends less on the dataset statistics.

B. Experiment setup

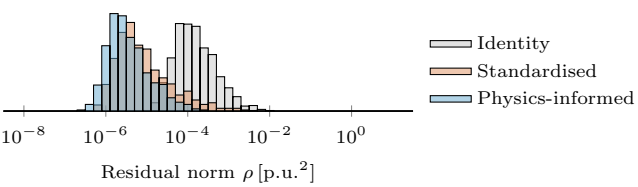
We train small fully-connected neural networks to predict power flow solutions following the residual formulation of [35]. The IEEE 9-bus system serves as an interpretable example, and the IEEE 14-, 30-, and 57-bus systems show that the trends carry to larger systems. As the aim is to isolate the effect of the non-dimensionalisation rather than to identify the best architecture or training procedure, we vary only the non-dimensionalisation scheme and keep the architecture, weight initialisation, training procedure, and training data identical, repeating each experiment over five random seeds against a fixed test dataset per system size. The setup is detailed in the Appendix, and the code is available at https://github.com/jbesty/powerup-2026-nondimensionalisation.

IV. Results

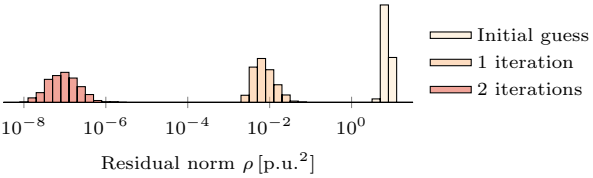
This section demonstrates the effect of non-dimensionalisation of the learning problem. We discuss the importance of the residual norm and its relation to the training loss. We then evaluate element-wise errors and the robustness under different training conditions. Additional experiments can be found in the Appendix.

A. Residual norm distributions

To analyse the performance of a solver, we evaluate the residual norm ρ of the solver on the test dataset. The resulting distribution gives an indication on the overall accuracy. Figure 1a displays the distribution of the residual norm for the three non-dimensionalisation schemes and shows a clear improvement from using the identity scheme to standardising to using the proposed physics-informed non-dimensionalisation. As a reference, we also plot the residual norm distribution of a Newton-Raphson-based solver on the test dataset initialised with a flat start, i.e., $V_i = 1.0\,\mathrm{p.u.}$ and $\varphi_i = 0.0\,\mathrm{rad}$. While Fig. 1b shows that the initial values have high errors, already two iterations reach a residual norm $\rho < 10^{-6}\ \mathrm{p\,u}^2$, quickly converging to the desired tolerance. While neural solvers can generally not be expected to reach such numerical tolerances, simply applying the proposed physics-informed non-dimensionalisation still yields improvements.



(a) Neural solver trained with different non-dimensionalisation strategies.



(b) Newton–Raphson solver after successive iterations.

Fig. 1: Distribution of the residual norm ρ on the test dataset.

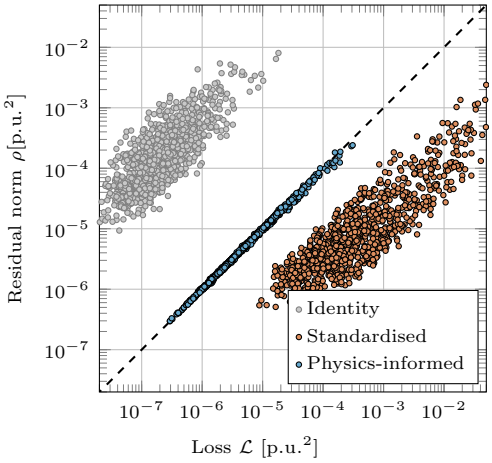


Fig. 2: Correlation between the training loss $\mathcal{L}$ and the residual norm ρ evaluated on the test dataset. The physics-informed non-dimensionalisation induces an almost perfect linear relationship in log–log scale.

B. Relation between the loss function and the residual norm

Due to the importance of the residual norm, we analyse how well the training objective $\mathcal{L}$ under each non-dimensionalisation correlates with the residual norm ρ. Figure 2 plots for each data point in the test dataset the loss value $\mathcal{L}$ on the x-axis and the residual norm ρ on the y-axis, both in logarithmic scale. The dashed diagonal line indicates where the loss and residual norm are equal. For the physics-informed non-dimensionalisation, the data points align closely with the diagonal line, so the loss of each sample is a reliable proxy for the residual norm: the training objective aligns with the objective pursued in numerical methods, and the loss value becomes directly interpretable[1]. In contrast, the other non-dimensionalisation schemes show a much weaker correlation between the loss and the residual norm. Hence, the loss value does not directly track the residual norm.

C. Comparison of the element-wise errors

While the residual norm ρ is the central metric—evaluated per sample or in its distribution—physical interpretability stems from understanding the voltage errors in the physical units and how the non-dimensionalisation affects these voltage errors. To analyse the voltage errors, we run multiple training runs of different training datasets, each with $|\mathcal{D}_{\text{train}}| = 250$, and random weight initialisations $\theta^{(0)}$. For each run, we test the non-dimensionalisation approaches.

To analyse the error distribution, we sort the absolute voltage errors of each variable to compute the respective quantiles. This allows us to compare the non-dimensionalisation approaches across the entire error distribution, rather than only comparing the mean or median errors. As an exemplary case, Fig. 3 shows the error quantiles of voltage magnitude V_1, one line for each training run. We observe that the median error

[1]As this dataset implies $\rho = 0$ for all samples, the residual norm ρ coincides with the loss $\mathcal{L}$; for cases where $\rho \neq 0$, the loss would track the difference between the exact and estimated residual normρ.

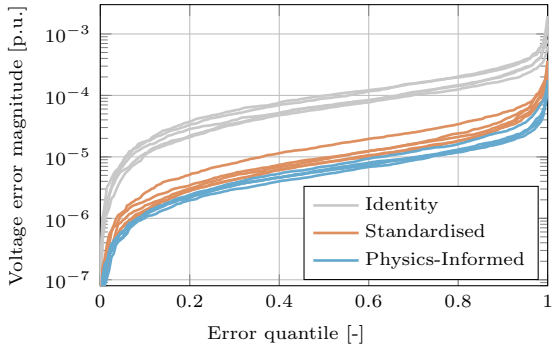


Fig. 3: Error quantiles across five training runs with different random initialisations. For the shown V_1, physics-informed non-dimensionalisation yields mostly the lowest errors.

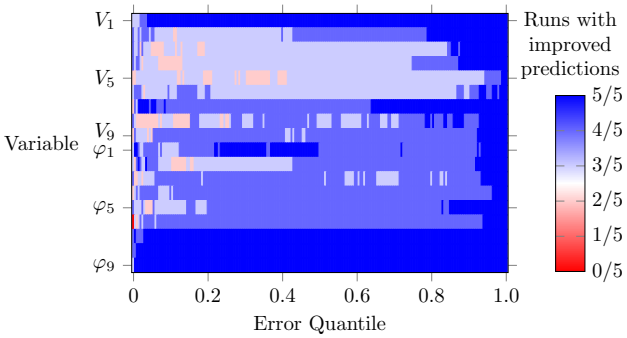
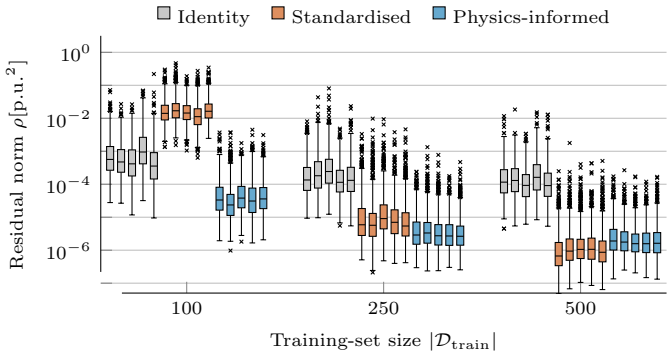
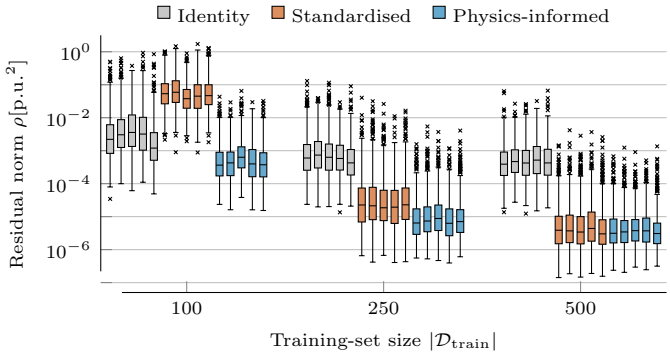


Fig. 4: Per-variable share of runs for which the physics-informed non-dimensionalisation yields smaller errors than standardisation, evaluated across error quantiles. Blue fields indicate improved performance of the proposed approach.

(quantile 0.5) lies around 1×10^{-4} p.u. for the identity non-dimensionalisation, and around 1×10^{-5} p.u. for the standardised and physics-informed non-dimensionalisation. The maximum error (quantile 1.0) shows the tail behaviour with approximately 10-50 times higher errors than the median. For the chosen variable, the physics-informed non-dimensionalisation yields the lowest errors across the runs and quantiles. For an overall assessment, we compare the error quantiles of the standardised and physics-informed non-dimensionalisation given the exact same training setup including the parameter initialisation and training dataset. Figure 4 shows the share of instances in which the proposed physics-informed non-dimensionalisation leads to smaller error than the commonly used standardisation. The blue fields mean that the physics-informed non-dimensionalisation yields smaller errors than standardisation, while the red fields indicate the opposite. We observe, that the physics-informed non-dimensionalisation performs better across nearly all quantiles and variables compared to the standardisation for the given setup. The improved performance indicated by the residual norm distribution in Fig. 1 is also reflected in the element-wise errors.



(a) Narrow distribution.



(b) Wide distribution.

Fig. 5: Residual norm distributions for varying training-set sizes on the (a) narrow and (b) wide distributions. The wider input domain raises the residual norms overall. For per-variable performance evaluation, see Figs. 4 and 6.

D. Robustness analysis

We test the performance robustness of the physics-informed non-dimensionalisation by altering different parameters, namely training dataset size, the dataset distribution characteristics, the power system size, and the neural network representation capacity. We highlight the most important findings here, further details can be found in the Appendix.

1) Dataset size: We vary the number of training samples $|\mathcal{D}_{\text{train}}|$ from 100 to 500 and evaluate the residual norm distribution on the test dataset across five training runs. The results are shown in Fig. 5a. As expected, more data boosts performance, however, the physics-informed non-dimensionalisation is least dependent on dataset size and shows already good performance on the smallest dataset.

2) Dataset distribution: Another critical factor is the distribution according to which the dataset is generated. While Fig. 5a was trained and tested on a *narrower* distribution, Fig. 5b shows the results for a *wider* distribution. The two panels differ little, the *wider* distribution overall produces larger residual norms, in particular with respect to the outliers. This behaviour is expected as the input domain $\mathcal{X}$ covers a larger area. If we consider, however, the performance in terms of the element-wise errors, we observe in Fig. 6 that the physics-informed non-dimensionalisation performs worse than standardisation for a much larger part of the variables and

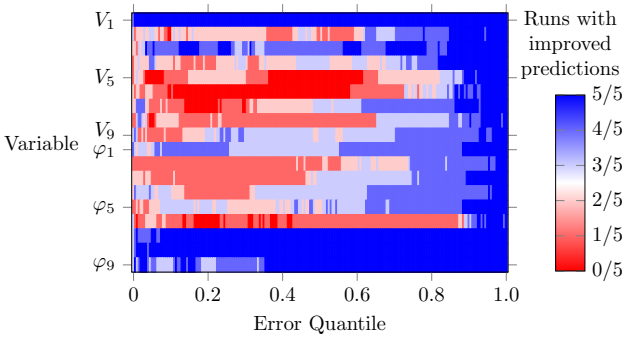


Fig. 6: Error ranking for the wide dataset at $|\mathcal{D}_{\text{train}}| = 250$. Under the wider distribution, standardisation yields smaller errors for many variables and quantiles—the multi-objective trade-off—while the physics-informed scheme still favours the most sensitive variables such as V_1 and the error tails.

quantiles. This reflects the multi-objective nature of the problem: the physics-informed scheme targets the residual norm, whereas standardisation spreads accuracy across variables, so neither dominates everywhere. Nonetheless, the maximum absolute errors and highly sensitive variables such as V_1 are still better approximated with the physics-informed non-dimensionalisation, which is consistent with the design of the approach.

3) System size and neural network capacity: In additional tests, see Appendix, we increase the system size and the neural network representation capacity. The results show that the physics-informed non-dimensionalisation improves performance most, when the learning is prone to overfitting, which occurs when there is little data or the model is too complex. In these cases, the physics-informed non-dimensionalisation provides a training objective, which improves the (in-distribution) generalisation from training to test data.

V. Conclusion

We have shown that the choice of non-dimensionalisation plays a central role in learning neural solvers for physical systems. By aligning the learning objective with the residual-based optimisation used in numerical methods, we derive a physics-informed non-dimensionalisation that provides an interpretable and physically meaningful assessment of prediction errors. The proposed approach aligns the training loss and numerical error metrics, reduces dependence on dataset statistics, and enhances robustness with respect to limited training data. In this way, error assessment for neural solvers is brought closer to established engineering practice.

The proposed non-dimensionalisation follows directly from the residual formulation of algebraic constraints. Extending the approach to differential equations and time-dependent systems constitutes an important direction for future work. Beyond individual systems, we see potential for advancing the notion of non-dimensionalisation across physical models in power systems. Such formalisation could improve the comparability of neural solvers and inform dataset sampling design and learning strategies for reliable neural solvers.

AI Usage Disclosure

The first draft of this manuscript was written without the use of AI. Subsequent iterations using AI involved restructuring, harmonisation of language, and improvements of clarity throughout the document. For plotting and writing code, AI has been used in form of autocompletion and debugging. The published repository has been refactor using AI.

Appendix

The public repository https://github.com/jbesty/powerup-2026-nondimensionalisation contains the full code, configurations, and results to reproduce the case study.

A. Power flow model

The physical systems considered in this case study are IEEE test cases described using a residual power flow formulation [35]. The input variables $\boldsymbol{u} \in \mathbb{R}^m$ are the control variables, namely generator power and voltage set-points as well as load power set-points. The outputs are the voltage variables $\boldsymbol{v} \in \mathbb{R}^{|\mathcal{N}|+|\mathcal{B}|}$, consisting of the bus voltage magnitudes V_i and branch angles φ_i. The algebraic constraints $\boldsymbol{F}(\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{|\mathcal{N}|+|\mathcal{B}|+1}$ are derived from Kirchhoff's laws. The system is over-determined, but for AC-feasible power flows the residual norm satisfies $\rho(\boldsymbol{u}, \boldsymbol{v}) = 0$.

The main system of analysis is the 9-bus system, while the larger systems (the 14-, 30-, and 57-bus cases) are used for the scalability analysis in Section D. All system parameters are loaded through PowSyBl [36].

B. Dataset sampling

The datasets are generated by sampling the input variables $\boldsymbol{u}$. We first sample the total apparent power consumption $S_{total} \in [\underline{S}_{total}, \bar{S}_{total}]$, and then distribute S_{total} uniformly random across all loads. Active and reactive load powers are obtained using power factor $\psi_k \in [0.9, 1.0]$ independently sampled for each load. Generator power set-points are sampled via uniformly sampling participation shares, and the voltage set-points are drawn independently from $V_{ref,k} \in [\underline{V}_{ref,k}, \bar{V}_{ref,k}]$ p.u.. To ensure AC-feasibility for all samples, a distributed slack is used.

For the 9-bus system, we consider dataset variants: a *narrow* and a *wide* distribution, defined by the parameter bounds in Table II, the latter intentionally increases variability in the inputs and outputs. The larger systems follow the identical sampling procedure; only the per-system power and voltage bounds differ.

Dataset	$\underline{S}_{total}$	$\bar{S}_{total}$	$\underline{V}_{ref,k}$	$\bar{V}_{ref,k}$
Narrow	1.5 p.u.	2.5 p.u.	1.02 p.u.	1.03 p.u.
Wide	1.0 p.u.	3.0 p.u.	1.00 p.u.	1.05 p.u.

TABLE II: Dataset sampling bounds.

C. Learning setup

The neural solver is implemented in Julia [37] as a feed-forward neural network with one hidden layer and 100 neurons (except for Fig. 8) and we use the hyperbolic tangent activation function. The trainable parameters θ are initialised using a Glorot-normal distribution [38].

The test dataset consists of $|\mathcal{D}_{\text{test}}| = 1000$ samples. For training, we use $|\mathcal{D}_{\text{train}}| \in [100, 250, 500, 1000, 2000]$ and set the validation set size to $|\mathcal{D}_{\text{validation}}| = 0.2\,|\mathcal{D}_{\text{train}}|$. Training is performed using the L-BFGS optimiser [39] for up to 10000 epochs, which provides robust convergence behaviour, while still being tractable for the small neural network.

D. Scalability behaviour

This section reports how the non-dimensionalisation behaves as the problem is scaled—in system size and in network capacity. The improved data efficiency can be related to the generalisation performance which we explore first, before reporting results on the network capacity and providing the full scale study results.

a) Generalisation: Figure 7 isolates the mechanism on the 57-bus system (single representative seed): for each scheme it plots the *mean* per-sample residual norm—the trained objective—on the training set (dashed) and test set (solid) against N, whose separation is the generalisation gap. Standardisation overfits at small N (at $N = 100$, mean train $\rho \sim 10^{-6}$ but test 0.36) and closes the gap only slowly. Physics-informed reaches the lowest test ρ at every budget and has all but closed its gap by $N = 2000$, i.e. it generalises from scarce data. Identity underfits, its two curves meeting at a high floor. The physics-informed transformation thus acts as an implicit conditioner, which is why its advantage is largest where data is most limited.

b) Network capacity.: To check that the advantage is not an artefact of one architecture, we repeat the study on the 30-bus system across five network sizes from 1×50 to 2×200

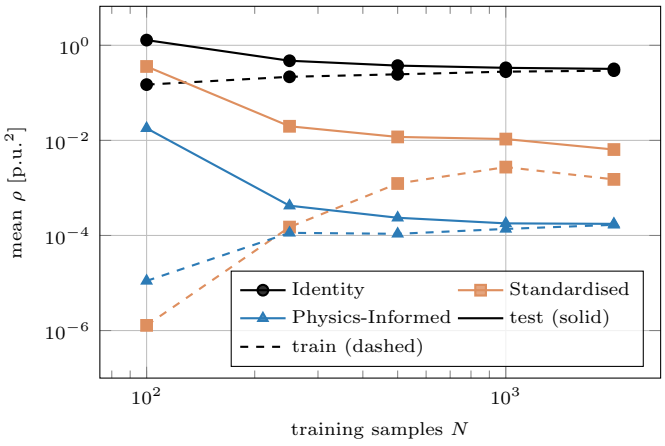


Fig. 7: Generalisation on the 57-bus system for a single representative seed: mean train (dashed) and mean test (solid) residual norm ρ versus N, per scheme. The per-sample mean is the trained objective, so the train–test gap is the generalisation gap, closing as N grows.

(the 30-bus system avoids compounding a larger network with a data-starved regime). Figure 8 shows the per-sample test-ρ distributions (15 boxes per panel, one per seed) at two dataset sizes, with the architectures ordered by increasing effective capacity. The physics-informed scheme is lowest at every capacity, and the *size* of its advantage grows with it: a too-small network cannot exploit the better-conditioned representation, so the schemes stay closer together. More data (bottom row, $N = 1000$) attenuates the trend but does not remove it, and the standardised / physics-informed ratio stays above unity for every architecture—so the advantage is a property of the non-dimensionalisation that a larger network only sharpens.

c) Full distributions.: For completeness, Figs. 8 and 9 show the full per-sample residual-norm distributions underlying these summaries—the capacity ablation and the system-size grid—confirming the trends at the level of individual seeds and samples.

REFERENCES

[1] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2002.

[2] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, 2nd ed. Cambridge University Press, Nov. 2008.

[3] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, ser. Classics in Applied Mathematics. Philadelphia, Pa: Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1996, no. 16.

[4] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Fourier Neural Operator for Parametric Partial Differential Equations," in *International Conference on Learning Representations*, 2021.

[5] S. Bai, V. Koltun, and J. Z. Kolter, "Neural Deep Equilibrium Solvers," in *International Conference on Learning Representations*, 2022.

[6] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs," *Journal of Machine Learning Research*, vol. 24, no. 89, pp. 1–97, 2023.

[7] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-Based Deep Learning," *Proceedings of the IEEE*, vol. 111, no. 5, pp. 465–499, May 2023.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY: Springer New York, 2009.

[9] K. Miettinen, *Nonlinear Multiobjective Optimization*, ser. International Series in Operations Research & Management Science, F. S. Hillier, Ed. Boston, MA: Springer US, 1998, vol. 12.

[10] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 2419–2426.

[11] I. Giagkiozis and P. Fleming, "Methods for multi-objective optimization: An analysis," *Information Sciences*, vol. 293, pp. 338–350, Feb. 2015.

[12] Y. Jin and B. Sendhoff, "Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 3, pp. 397–415, May 2008.

[13] O. Sener and V. Koltun, "Multi-Task Learning as Multi-Objective Optimization," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018.

[14] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 794–803.

[15] G. Montavon, G. B. Orr, and K.-R. Müller, Eds., *Neural Networks: Tricks of the Trade*, second edition ed., ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7700.

[16] I. Lagaris, A. Likas, and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, Sep. 1998.

[17] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, no. C, Nov. 2018.

[18] B. Donon, R. Clément, B. Donnot, A. Marot, I. Guyon, and M. Schoenauer, "Neural networks for power flow: Graph neural solver," *Electric Power Systems Research*, vol. 189, p. 106547, Dec. 2020.

[19] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, pp. A3055–A3081, Jan. 2021.

[20] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, "Characterizing possible failure modes in physics-informed neural networks," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 26 548–26 560.

[21] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, "An Expert's Guide to Training Physics-informed Neural Networks," Aug. 2023. [Online]. Available: https://arxiv.org/abs/2308.08468

[22] E. Buckingham, "On Physically Similar Systems; Illustrations of the Use of Dimensional Equations," *Physical Review*, vol. 4, no. 4, pp. 345–376, Oct. 1914.

[23] T. Szirtes, *Applied Dimensional Analysis and Modeling*. Elsevier, 2007.

[24] Y. Yuan and A. Lozano-Durán, "Dimensionless learning based on information," *Nature Communications*, vol. 16, no. 1, p. 9171, Oct. 2025.

[25] N. Lin, S. Orfanoudakis, N. O. Cardenas, J. S. Giraldo, and P. P. Vergara, "PowerFlowNet: Power flow approximation using message passing Graph Neural Networks," *International Journal of Electrical Power & Energy Systems*, vol. 160, p. 110112, Sep. 2024.

[26] A. Varbella, K. Amara, B. Gjorgiev, M. El-Assady, and G. Sansavini, "PowerGraph: A power grid benchmark dataset for graph neural networks," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, ser. NIPS '24, vol. 37. Red Hook, NY, USA: Curran Associates Inc., Jun. 2025, pp. 110 784–110 804.

[27] A. S. Zamzam and K. Baker, "Learning Optimal Solutions for Extremely Fast AC Optimal Power Flow," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Nov. 2020, pp. 1–6.

[28] P. L. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A learning method for optimization with hard constraints," in *International Conference on Learning Representations*, 2021.

[29] R. Nellikkath and S. Chatzivasileiadis, "Physics-Informed Neural Networks for AC Optimal Power Flow," *Electric Power Systems Research*, vol. 212, p. 108412, Nov. 2022.

[30] F. Fioretto, T. W. Mak, and P. Van Hentenryck, "Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 630–637, Apr. 2020.

[31] X. Pan, T. Zhao, M. Chen, and S. Zhang, "DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 1725–1735, May 2021.

[32] M. K. Singh, V. Kekatos, and G. B. Giannakis, "Learning to Solve the AC-OPF Using Sensitivity-Informed Deep Neural Networks," *IEEE Transactions on Power Systems*, vol. 37, no. 4, pp. 2833–2846, Jul. 2022.

[33] V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating Robustness of Neural Networks with Mixed Integer Programming," in *International Conference on Learning Representations*, Sep. 2018.

[34] S. Chevalier and S. Chatzivasileiadis, "Global Performance Guarantees for Neural Network Models of AC Power Flow," May 2024. [Online]. Available: http://arxiv.org/abs/2211.07125

[35] J. Stiasny and J. Cremer, "Residual Power Flow for Neural Solvers," Jan. 2026. [Online]. Available: http://arxiv.org/abs/2601.09533

[36] PowSyBl Contributors, "PowSyBl (Power System Blocks)," LF Energy, 2025.

[37] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, Jan. 2017.

[38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[39] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1-3, pp. 503–528, Aug. 1989.

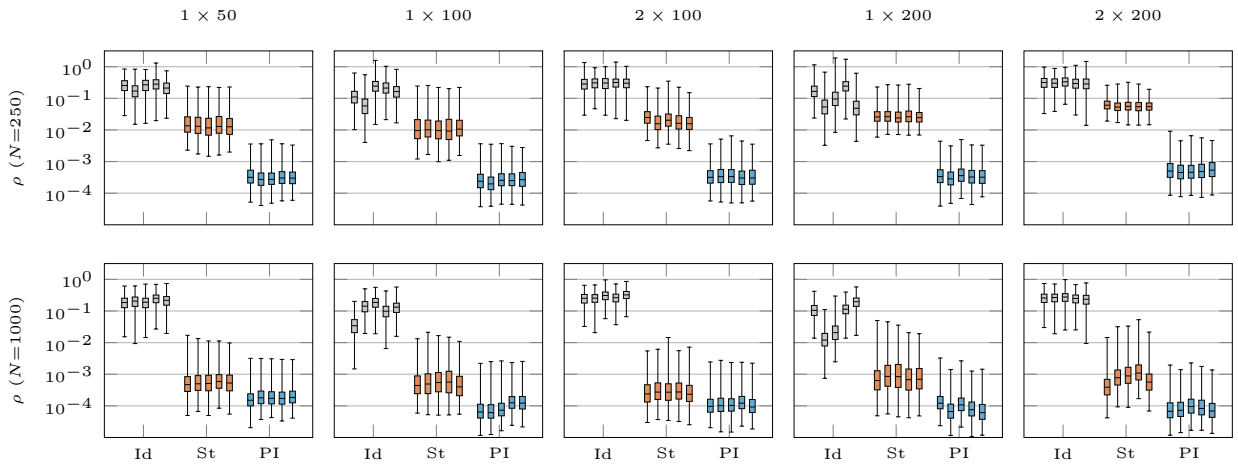


Fig. 8: Capacity ablation on the 30-bus system: per-sample test-ρ distributions across architecture (columns, ordered left to right by increasing effective capacity) at $N = 250$ (top) and $N = 1000$ (bottom). Each panel holds 15 boxes—five Identity (Id), five Standardised (St), five Physics-Informed (PI), one per seed; whiskers reach the minimum/maximum sample, with the shared ρ axis capped just above $\rho = 1$. The physics-informed advantage is present at every capacity and widens as capacity grows.

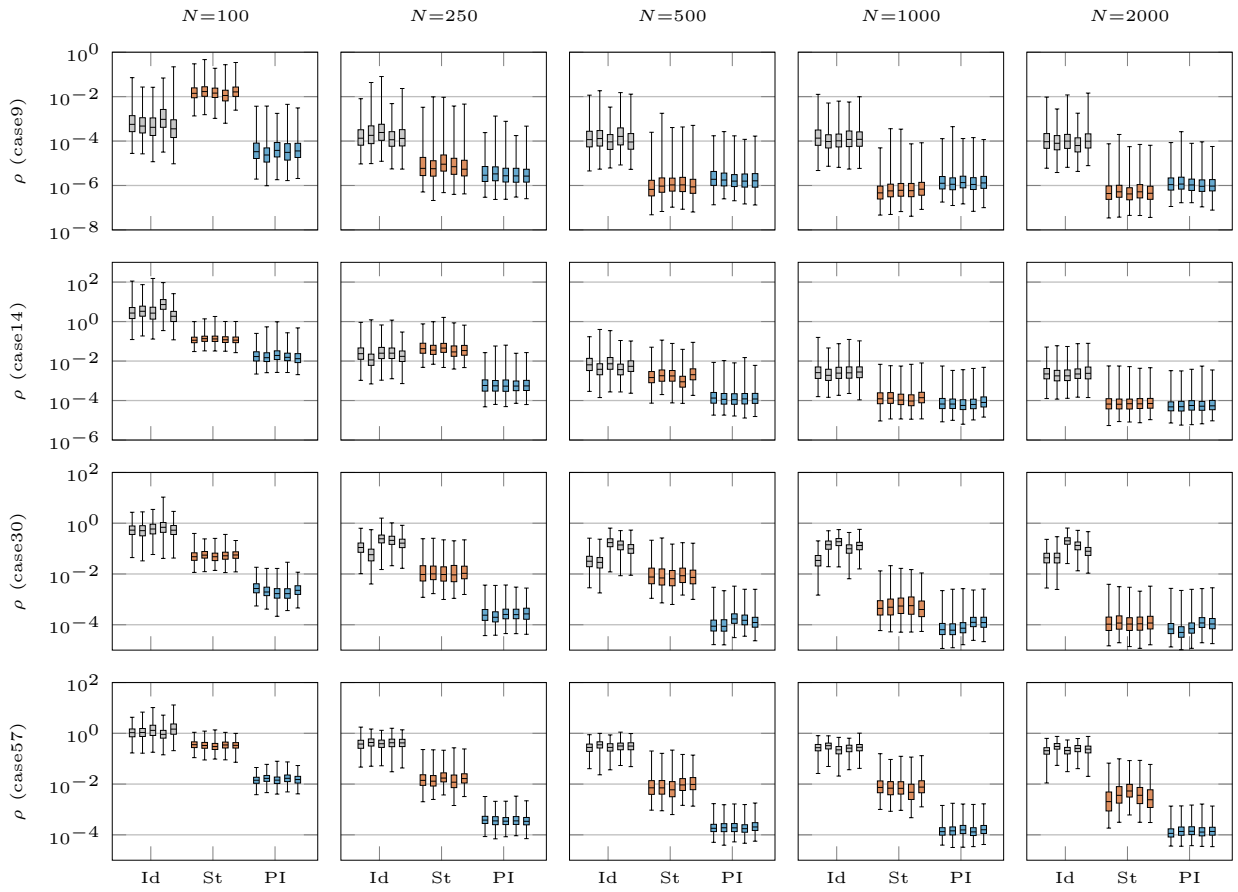


Fig. 9: Per-sample test-ρ distributions across system size (rows, 9–57 buses) and training-set size (columns, $N = 100$–2000). Each cell holds 15 boxes—five Identity (Id), five Standardised (St), five Physics-Informed (PI), one per seed—of the per-sample residual norm over the test set; whiskers reach the minimum/maximum sample. The ρ axis is shared within a row, with per-system limits. Physics-informed stays lowest with the shortest tails across the entire grid; the gap is largest at small N and on the larger systems.

I. Review #24A

A. Paper summary

This manuscript proposes a physics-informed non-dimensionalisation strategy that makes training-error assessment in machine-learning solvers consistent with the residuals used in classical numerical methods. The authors motivate why such scaling is important when multiple state variables with different units are aggregated into a single loss. A case study on the IEEE 9-bus system demonstrates the approach and shows a markedly tighter distribution of the residual norm.

B. Comments for authors

Strengths

Thank you for this clear and well-structured manuscript. I particularly appreciated the early motivation and framing, which are easy to follow and make the contribution intuitive. The paper addresses a timely challenge and offers a practical strategy to reduce errors in neural solvers. One question I had is how the proposed approach scales to larger test systems and/or more diverse learning tasks. The results for the narrow vs. wider operating ranges suggest that gains may diminish as data variability increases, while the overall error rises. Additionally, a brief discussion of expected scaling behaviour and limitations would strengthen the paper.

Weaknesses, Suggestions and Questions

Here are a few questions and suggestions to further improve clarity and presentation:

1) In a few places (e.g., IV.B) ρ is used as a subject. Using "residual norm" would be clearer.
2) How was the final neural-network architecture selected (e.g., any hyperparameter tuning/ablation)? Section III.C describes the setup, but the rationale for choosing it is not yet clear.
3) In the results section, it was sometimes hard to track which of the three approaches is being used and which equations/settings they correspond to. A summary (end of Methods or start of Results) listing the three approaches and the associated equations/setup would help.
4) The intuitive explanation for why non-dimensionalisation is needed is very helpful. If possible, I would appreciate a similarly intuitive, high-level explanation that compares the three non-dimensionalisation strategies.
5) Minor formatting: in Section III.B, it seems one occurrence of $\underline{S}_{\text{total}}$ should not be bold.
6) Regarding the dataset sampling: does each sample use a single (shared) power factor across all loads, or is the power factor varied per load within a sample?
7) Several figure discussions would benefit from first describing what the figure shows (axes, groups, and key patterns) before concluding. A few concrete examples:
   a. Fig. 2: the narrative moves between approaches in a way that is a bit hard to follow. For example, after discussing the physics-informed approach, the text states that the others have no clear patterns (to me they appear to occupy distinct regions), and then refers again to a "strong matching." Which approach is referenced at each point? And why are the other distinct regions not considered as patterns?
   b. Fig. 5: the text begins with the conclusion. I would consider first summarising what the figure shows, then interpreting it to make it easier for a reader to follow the argument.
   c. Fig. 3 is currently referenced only briefly. It could be used more explicitly to define and illustrate the error quantiles and thereby motivate Fig. 4. Relatedly, a short formal definition of "error quantiles" would be helpful.
   d. Fig. 4 was somewhat unintuitive for me at first. A brief explanation of (i) the error-quantile concept, (ii) what is stacked on the y-axis, and (iii) what the colours represent would improve accessibility. In the caption, "values above 0.5 indicate improved performance" seems to refer to the colour scale; stating this explicitly in both the caption and the text would avoid ambiguity.
8) Section IV.C mentions that the non-dimensionalisation approaches are tested for each run. What are the associated data splits and how are test metrics computed?

C. Summarize your recommendation in one to two sentences

The paper is clear, well motivated, and provides a useful physics-informed non-dimensionalisation strategy that improves residual-based error assessment for neural solvers. I recommend a few clarifications: improve the mapping between methods and results, especially in the figure discussion, and briefly discuss expected scalability and limitations.

— Johanna Vorwerk

II. Review #24B

A. Paper summary

This paper looks at deficiencies in the assessment of neural solvers' accuracy (specifically with neural solvers targeted towards steady state power systems problems) and develops a construction which produces a training objective more akin to conventional numerical solvers. Results are shown on the IEEE 9-bus network for a model which learns power flow solutions.

B. Comments for authors

Strengths:

- I greatly appreciate the author(s)' perspective of trying to formalize more of the learning-for-OPF space / neural solver space. Looking at neural solvers from the lens of conventional / numerical solvers is a good perspective to have since many of us are familiar with assessment metrics (convergence rate, constraint violation tolerance, optimality tolerance, etc.) used in conventional solvers.
- I know I left a lot of bullets under the "weaknesses" column, but I do believe there's merit in this paper and the general concept, with a bit of re-writing and a larger/harder test case.

Weaknesses:

- The paper should really briefly define "non-dimensionalism" in this context, especially since it is mentioned in the title.
- The authors are perhaps conflating the general concept of "neural solvers" with only networks with explicit layers. They do not discuss implicit layers at all, which enforce a different kind of input-output relationships within a deep learning model often by using conventional solvers in the forward pass. This important body of work (see any recent model that regards hard constraint satisfaction - e.g. DC3 by Donti et. al). the authors should specify that they are only looking at a subset of neural solvers and how the thesis does not necessarily carry to other models.
- The numerical experiments being performed on the IEEE 9-bus system is...quite limiting with respect to illustrating strong results to the reader. This is a very small system and certainly not the focus of papers which train neural networks to learn AC OPF solutions (i.e. a very high accuracy for solving this problem could be achieved without the "physics-informed" part of the loss function, and beyond that, a conventional solver would take less than a second on a laptop computer). I understand the results are illustrative, but I am struggling to see the practical use with a 9-bus network where the loads are only perturbed +/- 10% from the base loading value. It additionally looks like the application is just AC PF, not OPF, whereas most papers in the literature tackle the harder OPF problem.
- It is really not clear what "Identity" and "Standardised" are in the results. There needs to be an explicit section that discusses these. In general, maybe page limit was an issue here, but there needs to be more "setting the stage" during the explanations/writing of the paper. It makes it a little hard to follow, which limits the probability that people will use these new metrics. It reads a little bit like when someone has been working extremely hard on a problem and is in the weeds with the technical details, and writes well about the technical details, but needs to zoom way out and remember that it is the reader's first time even considering this problem.
- Minor comment, but since there is plenty of space, since the paper is highly focused on learning models for OPF, and it's a power systems conference, there should be more references focused on learning-for-OPF and the like. It is quite limited and there is such a wide berth of papers in this area at this point.

C. Summarize your recommendation in one to two sentences

They only illustrate results on a 9-bus system using AC power flow, whereas the application area for this analysis are papers developing models for thousand-bus AC optimal power flow problems. I don't know how I can be convinced with such a limiting and small test case.

— Anonymous reviewer

III. Review #24C

A. Paper summary

This paper proposes a physics-informed non-dimensionalisation strategy for neural solvers of physical systems addressing an important problem of scaling multiple physical variables at different scales and with different units. The authors derive scaling from the algebraic residual formulation and demonstrate its use in the supervised training loss. They also show how this loss aligns with the residual norm used in numerical solvers. Authors argue that this reduces dependence on dataset statistics, yields interpretable training objectives, and improves robustness under limited training data. The method is demonstrated on an IEEE 9-bus system.

B. Comments for authors

I find the topic very relevant, well motivated, and the presented ideas to be clear and intuitive. The paper is well written with a clear exposition of the problem setup and experiments.

1) The case study with the IEEE 9-bus system represents a simpler setting. For future work it would be interesting how the method tackles large scale problems emerging in real-world systems.
2) The paper would benefit from a clear distinction from physics-informed neural networks upfront.
3) the method depents on choosing residual weighting matrix W_r, but does not provide specific algorithmic procedure or empirical guidance how to choose this matrix in practice. In this paper, the authors choose W_r to be identity, but I believe this choice was not clearly articulated.

4) The paper would benefit from releasing code to enable reproducibility and wider evaluation by the community.
5) Limitations to algebraic systems. While algebraic power-flow is an important use case, many power-system problems are dynamic in nature modeled either by network ODEs or DAEs. The paper would benefit from a dicsussion on how could this method be adopted in such settings?

C. Summarize your recommendation in one to two sentences

I think this is an interesting preliminary paper that fits well in the PowerUP conference. It would be interesting to see how the method develops to tackle large scale problems involving differential equations.

— *Jan Drgona*

IV. Brief response to reviewers (optional)

We sincerely thank the reviewers for feedback, questions, and suggestions. Most of the comments are addressed directly in the manuscript and itemised in the accompanying list of changes. Here, we would like to respond to a few points that are more conceptual.

System size. We agree that the scaling behaviour is a central question, which we have investigated with experiments up to a 57-bus system. From a computational point of view, we believe scalability is well predictable, as the entire scheme only requires the calculation of the Hessian (and its inverse) across the dataset (or part of it). After this pre-computation, the training itself requires no additional resources. From the effectiveness perspective, the additional experiments are encouraging: the physics-informed non-dimensionalisation becomes even more effective with increasing system size. We seem to require fewer data points, compared to the other schemes, to achieve good in-distribution generalisation.

Scope: power flow first, OPF and dynamics as future steps. This work deliberately targets the AC power flow residual, as it directly provides the algebraic formulation in (1) and has a clear numerical solution equivalent. Dynamics are potentially adaptable if they are expressed in a similar algebraic form, e.g., by applying the trapezoidal integration scheme. Regarding AC OPF, the additional inequality constraints and the cost objective have to also be considered in the multi-objective problem. If they were formulated as residual functions, the question of the relative weighting against each other becomes important. This forces a deliberate design of W_r which we could avoid for the simple power flow, as all residuals are expressed as currents and in the same units, hence the simple choice $W_r = I$ here. Nonetheless, dynamics and OPF are relevant directions for future work.

Explicit- vs implicit-layer solvers. Implicit solvers are indeed an interesting direction. We believe the formulation of (1)-(3) does not exclude such methods; a corresponding implementation could be insightful.

V. Brief summary of changes made to the final paper

We thank the reviewers for their comments. Below we list the main changes, keyed to the points raised.

Introduction.

- Elaborated on the idea of non-dimensionalisation.
- Included a distinction from PINNs.
- Related the work to learning for AC-OPF.

Methodology.

- Corrected the input transform. In the submitted version, the affine transform inverted the input weighting ($A_x^\top A_x = W_x$), scaling high-sensitivity inputs the wrong way. Section II.E and Table I now use the consistent form $A_x^\top A_x = W_x^{-1}$, $A_y^\top A_y = W_y^{-1}$, matching the backward-error interpretation; all results are regenerated accordingly.
- Stated that $W_r = I$ is a deliberate modelling choice, open to problem-specific adaptation.

Case study.

- Added the IEEE 14-, 30-, and 57-bus systems alongside the 9-bus case to show that the trends scale. Description and results primarily in the Appendix, the Case Study Section provides the overview necessary for the results.
- Added Table I summarising the three schemes (Identity / Standardised / Physics-informed) with an intuitive comparison.
- Specified the sampling: power factor drawn independently per load, load split per sample, distributed slack ensuring AC-feasible targets.
- Specified the protocol: fixed test set (1000 samples), validation set= $0.2|\mathcal{D}_{\text{train}}|$, five seeds, metrics on the held-out test set.
- Justified the architecture (one hidden layer, 100 tanh units) with a capacity ablation across five network sizes.
- Released the code (repository linked in the paper).

Results.

- Rewrote the figure discussions to describe axes and groups before drawing conclusions.
- Explained the error quantiles more explicitly and used them to motivate the ranking heatmap, whose stacked axis and colour scale are now explained in text and caption.

- Extended the robustness study to system size and network capacity (details in the Appendix), and reframed the wide-distribution result as a multi-objective trade-off: the physics-informed scheme targets the residual norm and the most sensitive variables, standardisation spreads accuracy across variables.

Minor. Replaced ρ as a sentence subject with ”the residual norm”; removed the stray bold face on $\overline{S}_{\text{total}}$