

Study Design and Demystification of Physics Informed Neural Networks for Power Flow Simulation

Milad Leyli-abadi¹ , Antoine Marot² , and Jérôme Picault² 

¹ IRT SystemX, Palaiseau, France

² RTE, Paris, France

milad.leyli-abadi@irt-systemx.fr

Abstract. In the context of the energy transition, with increasing integration of renewable sources and cross-border electricity exchanges, power grids are encountering greater uncertainty and operational risk. Maintaining grid stability under varying conditions is a complex task, and power flow simulators are commonly used to support operators by evaluating potential actions before implementation. However, traditional physical solvers, while accurate, are often too slow for near real-time use. Machine learning models have emerged as fast surrogates, and to improve their adherence to physical laws (e.g., Kirchhoff's laws), they are often trained with embedded constraints—known also as physics-informed or hybrid models. This paper presents an ablation study to demystify hybridization strategies, ranging from incorporating physical constraints as regularization terms or unsupervised losses, and exploring model architectures from simple multilayer perceptrons to advanced graph-based networks enabling the direct optimization of physics equations. Using our custom benchmarking pipeline for hybrid models called LIPS, we evaluate these models across four dimensions: accuracy, physical compliance, industrial readiness, and out-of-distribution generalization. The results highlight how integrating physical knowledge impacts performance across these criteria. All the implementations are reproducible and provided in the corresponding Github page³.

Keywords: Power flow simulation · physics informed machine learning · comprehensive benchmark.

1 Introduction

To enable the power grid management and ensure the stability, the grid operators should continuously monitor the infrastructures and suggest remedial actions when necessary. However, such actions cannot be applied directly on the grid without any prior impact analysis as they may cause damages in the worst-case scenarios. The integration of renewable energy sources, e.g., wind and solar, further increased the uncertainty level and required the power systems to be flexible [1]. Hence, power flow simulations are extensively used to help grid operators in their decision-making process. The simulations allows to study different scenarios or solutions and simulate

³ <https://github.com/Mleyliabadi/pinns-powergrid>

the impact of their actions in a digital environment, before their application on the real grid.

In the real-world scenarios, thousands of these simulations may be required, to avoid any nonlinear effect of some local changes. The power flow simulations are currently based on the resolution of the physical equations which are generally solved using iterative numerical optimization methods such as Newton-Raphson [19]. Although these methods have the advantage of being precise and compliant with respect to physical constraints, their exponential time complexity does not allow their use in a large-scaled grid.

In this context, various studies have been interested in the use of machine learning-based approaches [4]. They have also been used for power flow computation [5, 8]. The graph neural networks are also successfully used for this purpose [3, 14]. Their fast inference time and leveraging the properties derived from graph theories may be a key-enabler to replace the complex physics-based solvers. Among the disadvantages of these approaches, we may cite their generalization problems (e.g., grid scale-up and unseen data), physics noncompliance and to be dependent on the quality and the volume of the data.

Recently, the physics-informed neural networks (PINNs) [16] are successfully applied in various domains to solve physics-based problems such as fluid dynamics [20], scientific simulations [9] and weather forecasting [11]. Various similar works are also conducted for power flow simulations [6, 12, 15]. As such, the physical constraints may be leveraged in different ways to make the data-driven approaches more compliant to the physical context. Among the most used strategies we can cite the direct use of the physics equations as the optimization objective during the training [6], warm start points using machine learning followed by optimization [2, 22], or their use as a regularization term. However, each of these strategies comes with their pros and cons. Most of the studies use one or another, but do not report and compare their results using a set of standardized and comprehensive evaluation criteria.

In this paper, we try to bridge this gap; that is, to design a set of step-by-step studies to show the impacts of considering physical constraints whether as a regularization term in a simple neural network (MLP), or by using a two-stage approach where the warm start points inferred by an MLP are fed to a message passing graph-based approach optimizing directly the physics equations. We also suggest evaluating performance using an extensive benchmark through various evaluation criteria categories based on our Learning Industrial Physical Simulation (LIPS) framework [13]. In addition to traditional machine learning metrics, using this standardized framework, we evaluate the performance with respect to physics compliance, out-of-distribution generalization and industrial readiness. All the studies provided in this paper are also available on the Github repository. All the datasets used for the training and evaluation of models are designed and generated for the sake of this study by considering real-world scenarios and could easily be reproduced using the provided scripts.

In different designed studies, we consider the steady-state hypothesis of observations by considering various grid configurations. These configurations may reflect the actions that the grid operators may take in different scenarios to keep the grids in a stable state. As this work aims to be also didactic and simple, we use a Direct Current

(DC) solver for the generation of different scenarios. We make use also of two different grid sizes for the experiments as a way to evaluate the scale-up capability of different benchmarked approaches.

The rest of the paper is organized as follows. Section 2 describes the general framework of the study including the power flow simulation problem and the technical details of the generated datasets. Section 3 provides the details of different studies, along with graphical visualizations, that are designed to solve the power flow problem whether using pure machine learning techniques or their physics-informed variants. Section 4 presents the experimental settings and the corresponding benchmark results. Finally, Section 5 concludes the paper by highlighting insights for future works.

2 Problem framework and datasets

As mentioned, to operate the power grid in near real-time, a high number of power flow simulations are required by the operators to study the impact of the potential remedial actions. In addition to the industrial requirements such as fast simulations and scale-up capabilities, these simulations should be representative of real-world power grid dynamics, i.e., they should be reliable in various configurations where the data distribution could be slightly changed. For the models to be compliant to physical laws, they should consider them as constraints during the optimization. The next section presents the technical details of physical solvers which are the mostly used tools for simulations of power flows. It follows by an introduction of the datasets that are generated for the purpose of this study. Table 1 presents the notations that are used throughout the paper.

2.1 Physical solvers

Physical solvers which are currently used for the grid simulations are based on the resolution of power flow equations [18,21]. These equations for the alternative current (AC) are given by:

$$\begin{cases} 0 = -p_k + \sum_{m=1}^K |v_k| |v_m| (g_{k,m} \cdot \cos(\theta_k - \theta_m) + b_{k,m} \sin(\theta_k - \theta_m)) & \text{Active power;} \\ 0 = q_k + \sum_{m=1}^K |v_k| |v_m| (g_{k,m} \cdot \sin(\theta_k - \theta_m) - b_{k,m} \cos(\theta_k - \theta_m)) & \text{Reactive power,} \end{cases} \quad (1)$$

where g_{km} and b_{km} are real and imaginary parts of the admittances connecting the nodes k and m . To solve these equations, an iterative optimization algorithms like Newton-Raphson may be used. Such iterative methods require a high number of steps to be converged and do not satisfy the near real-time requirement of power flow simulations. Hence, several contributions in state-of-the-art are interested in the use of physics-informed machine learning to cope with this problem. However, none of these papers provides an in-depth analysis of considering the physical knowledge when using machine learning algorithms based on neural networks.

To simplify the power flow simulation problem, in this paper we suggest to use the direct current (DC) power flow system which reduces the physical equations to:

$$0 = -p_k - \sum_{j=1, j \neq k}^K b_{kj} (\theta_k - \theta_j). \quad (2)$$

Table 1: Notation table

Symbol	Description
i	Index associated with an observation (sample)
k	Index associated with a grid node
ℓ	Index associated with a power line
or	Index associated with the "origin" extremity of a power line by convention
ex	Index associated with the second extremity of a power line by convention
p_k	Active power at node k
q_k	Reactive power at node k
v_k	Voltage at node k
θ_k	Voltage phase (phasor) at node k
p^ℓ	Active power flow over line l
q^ℓ	Reactive power flow over line l
a^ℓ	Current over line l
τ	Topology vector indicating the bus connectivity
L	Number of power lines in the grid
K	Number of substations (nodes) in the grid
B	Number of buses in the grid
N	Number of samples in training set
M	Number of samples in test set

The DC simplifies the AC equations by assuming small phasors (voltage angles) differences and implying constant bus voltages (1 p.u.). DC does not generate reactive power. This means that, in a DC system, only real (or active) power is transmitted. These make the DC solver significantly faster to compute in comparison to AC solvers.

2.2 Datasets

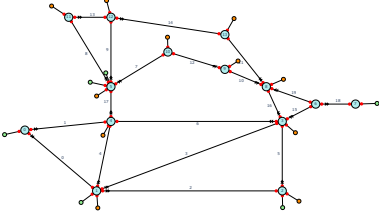
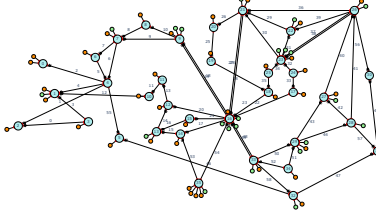
Two different power grids with different sizes are investigated in this paper. Their characteristics are presented in Table 2. The first ones includes a set of 14 nodes and 20 power lines. Due to its small size, the non-linear impact of topological changes propagates more severely throughout the grid. To gain insight into the scalability of the models, a slightly larger grid with 36 nodes and 59 power lines is also considered.

A specific scenario is designed for the generation of datasets using a DC solver and *lightsim2grid*⁴ Python package. In this scenario, we consider a set of reference topology which are randomly applied to each sample (observation). The reference topology may include the grid configurations where the elements connected to each substation are randomly connected to one of two bus bars at each node. It could also include the original observation without any changes (with a predefined probability). On top of this reference topology, we consider a set of additional operations including power line disconnections and further bus bar reconfigurations. In total, four different datasets are generated representing the following characteristics:

- *Training dataset*: it includes 100,000 samples. In addition to the reference topologies, sampled randomly, it may include at most one line disconnection;

⁴ <https://github.com/Grid2op/lightsim2grid>

Table 2: Two environments considered in this study along with their characteristics

Env name	IEEE 14	NeurIPS 2020
Grid		
# substations	14	36
# lines	20	58
# generators	6	22
# loads	11	37

- *Validation dataset*: it includes 10,000 samples and presents exactly the same distribution as the training set. This is used to analyze the over-fitting problem of ML-based approaches;
- *Test dataset*: it includes 10,000 samples. In addition to the randomly sampled reference topologies, it includes one random power line disconnection for each observation;
- *Out-of-distribution (OoD) dataset*: it includes 10,000 samples. It is considered as a secondary test data set to evaluate the generalization capability of the models on the observations represented by a slightly different distribution than the training set. In addition to reference topologies, it includes two simultaneous power lines disconnections which have never been considered in other datasets.

3 Study design

In this section, different neural network architectures along with their technical and theoretical details are described. We start with the simplest architecture without considering the physical constraints. It is followed by more complex architectures, where the physical constraints may be used as a regularization term or as the optimization objective. All the studies are accompanied by a schematic representation. In all these graphical representations, the first row presents the general steps, the corresponding visualization and architecture are shown in second row, and last row represents the details concerning the optimization objective (loss function).

Multi-layer perceptron (MLP) The first configuration consists in considering the power flow computation as a simple supervised regression problem. Figure 1 demonstrates the entire procedure for computing the power flow. As can be seen, the MLP takes the injections ($\mathbf{P}_{prod}, \mathbf{P}_{load}$) alongside the topology vector ($\boldsymbol{\tau}$) *i.e.*, the bus connectivity and line status as inputs, and predicts the phasors $\hat{\boldsymbol{\theta}} = MLP(\mathbf{P}_{prod}, \mathbf{P}_{load}, \boldsymbol{\tau})$ at the power lines. The active powers at the origin \mathbf{P}_{or} and extremity \mathbf{P}_{ex} sides of the power lines could be easily computed from the predicted phasors in a post-processing step. In this configuration, the loss function is simply the mean squared error between the predictions of the MLP and the ground-truth values.

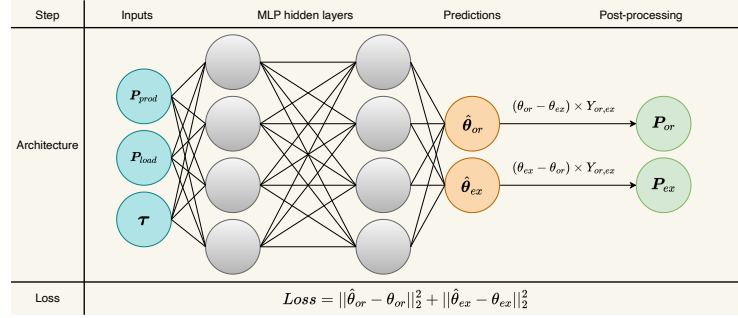


Fig. 1: Multi-layer perceptron (MLP). The inputs X (injection + topology) in blue , the model predictions $\hat{y} = (\hat{\theta}_{or}, \hat{\theta}_{ex})$ in orange (phasors) and post-processed outputs (P_{or}, P_{ex}) corresponding to active powers in green.

Message passing mechanism (MP) In this section, we provide the technical details concerning the message passing mechanism for power flow simulation, which enables the computation of physical constraint using graph theory. The classic message passing mechanism, as demonstrated in Figure 2, allows to propagate messages from the neighboring nodes. The propagation consists of three steps: message computation, aggregation, and update. By adding more layers (increasing the hops in the graph terminology), each message could include the information contained in farther nodes in the graph. This is shown by changing the nodes colors from light gray to green from the second hop neighbors.

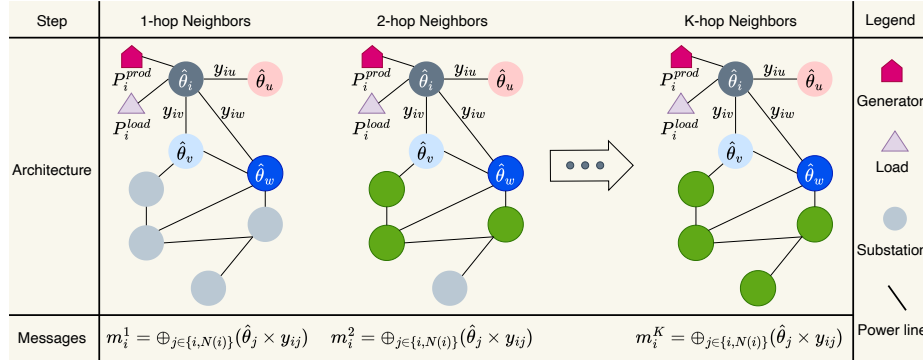


Fig. 2: Message-Passing (MP) mechanism demonstration. The message includes the information from farther nodes by adding extra layers (k -hop neighborhood). The maximum number of effective layers is equal to K which is equal to the number of nodes in the graph. In power grid context, each message represents the incoming power flow to a node i from a neighboring node j which is obtained by multiplying the phasor at node j with the admittance value on the power line connecting two nodes $(\theta_j \times y_{ij})$.

However, this may not be the required behavior for power flow computation, as we want the messages to represent the underlying physics equation. As shown in the last row of Figure 2, each message is an aggregation of the information contained in neighboring nodes multiplied by the admittances on power lines. Alternatively, it represents the power flow that entered a specific substation. In the following, we exploit this message-passing mechanism in two different ways:

1. *To compute the local conservation law.* This is very helpful, as it allows to compute the physical constraint, which could be exploited by the auto-differentiation of neural networks. In the backward step, it enables the estimation of neural network parameters with respect to the gradients of local conservation law. (see Figure 3);
2. *To update the phasors using physics equation.* The same message passing mechanism could be used to compute the new estimations of the phasors. This is used by graph neural solvers, explained in next sections.

Regularized MLP (MLP Reg) The regularized MLP adds a regularization term into the loss function of a classical MLP and adopts a slightly different architecture, as can be seen in Figure 3. It uses the same set of inputs as MLP, but to compute the local conservation error through the message-passing mechanism, it predicts the phasors at the bus level instead of the extremities of the power lines. The local conservation error LC_{error} obtained using a single message passing layer is considered as a regularization term in the loss formulation and represents the physical constraint that the model output should satisfy. Once the phasors are predicted, the active powers at the power line extremities could be easily computed at the post-processing step.

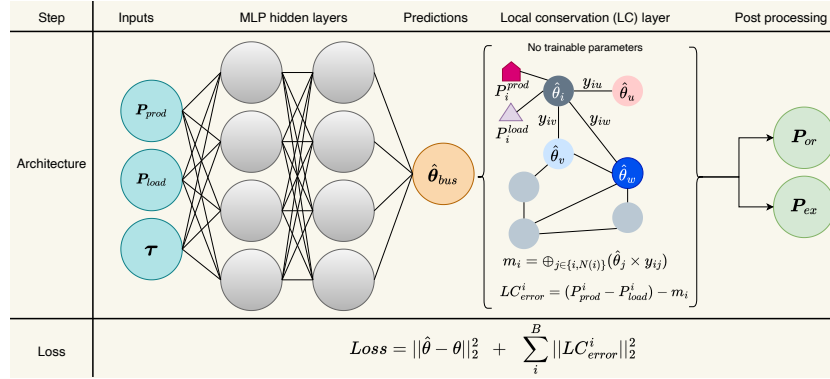


Fig. 3: Regularized MLP. The inputs X (injections + topology) in blue, the model predictions $\hat{y} = \{\hat{\theta}_{bus}\}_{bus \in (1, \dots, B)}$ in orange (phasors) and post-processed outputs (P_{or}, P_{ex}) corresponding to active powers in green. A local conservation (message passing) layer is added after the predictions to compute physical constraint error which is considered as a regularization term in the loss function.

Message-passing and physics constraint as optimization (MP Opt) In this specific study, we consider the message-passing mechanism for both updating the phasor values and to compute the local conservation error. As can be seen in Figure 4, the phasors are initialized with zeros (flat initialization). Next, the two message-passing layers to update the phasors and to compute local conservation are interleaved. In the following, we explain the theoretical details of the update step.

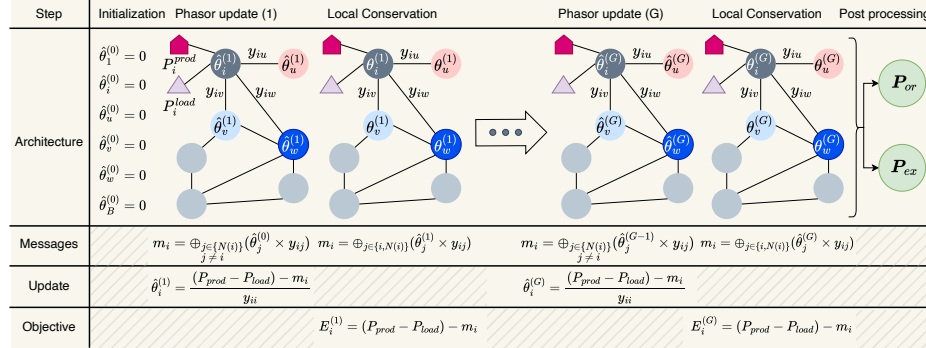


Fig. 4: Message-passing as physics optimization with flat initialization of phasors (MP Opt). The architecture consists in interleaving two message-passing layers to compute the new θ s (optimization) and local conservation error.

The computation of new phasor values over message-passing layers is based on the local conservation law formulation, which for a given substation i is given by:

$$p_i^{prod} - p_i^{load} = \sum_{\ell \in N(i)} p_i^\ell, \quad (3)$$

where p_i^ℓ designates the power flow at a power line ℓ connected to a substation i . This is equivalent to:

$$p_i^{prod} - p_i^{load} = \sum_{j \in \{i, N(i)\}} \theta_j \times y_{ij}, \quad (4)$$

where θ_j represents the phasor at a neighbor node j and y_{ij} is the admittance between two adjacent nodes i and j which is extracted from the admittance matrix Y . To compute the new phasor values at a node i , by considering $N(i) = \{u, v, w\}$ as its neighbors (see Figure 4), the Equation 4 becomes:

$$p_i^{prod} - p_i^{load} = (\theta_i \times y_{ii}) + \underbrace{(\theta_u \times y_{iu})}_{\text{message from node } u} + \underbrace{(\theta_v \times y_{iv})}_{\text{message from node } v} + \underbrace{(\theta_w \times y_{iw})}_{\text{message from node } w}, \quad (5)$$

where y_{ii} is the admittance at the node i . As can be seen, the messages computed to update the phasor include only the information contained in the neighboring nodes, and not the graph node for which the update should be computed. This could be managed using self-loops in message-passing mechanism. Finally, the new value of

our target, which is the phasor θ at node i and for a layer k , is computed as follows:

$$\theta_i^{(k)} = \frac{(p_i^{prod} - p_i^{load}) - [(\theta_u^{(k-1)} \times y_{iu}) + (\theta_v^{(k-1)} \times y_{iv}) + (\theta_w^{(k-1)} \times y_{iw})]}{y_{ii}}. \quad (6)$$

Finally, the local conservation error could easily be calculated for each layer k and at each node i from Equation (4) as:

$$E_i^{(k)} = p_i^{prod} - p_i^{load} - \sum_{j \in \{i, N(i)\}} \theta_j \times y_{ij}. \quad (7)$$

Physics-informed message-passing with warm initialization (PIMP) This last study consists in adding a learning paradigm into the optimization-based solution introduced in the previous study. As can be seen in Figure 5, instead of flat initialization, the phasors are initialized using an MLP neural network (warm initialization). We expect that this initialization, by introducing learnable parameters, would reduce the required number of iterations (message-passing layers) for convergence. As such, back-propagation takes into account the graph operations when updating the MLP parameters, which would allow the better initialization of phasors by considering implicitly the physical constraint. In contrast to studies that exploit the GNNs [14], our approach relies on the resolution of physics equations.

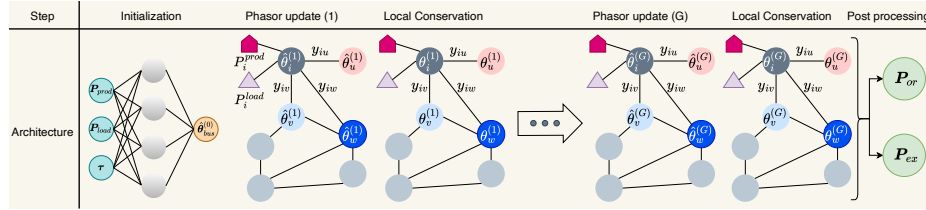


Fig. 5: Physics-informed message-passing (PIMP) with warm initialization ($\hat{\theta}^{(0)} = MLP(P_{prod}, P_{load}, \tau; \omega)$). The messages, updates and objectives are exactly the same as the one shown in Figure 4

4 Benchmark and evaluation

This section introduces the experimental setups used by each technique during the evaluation, details the benchmark testbed and the evaluation criteria and presents the obtained results.

4.1 Experimental settings

The Direct Current (DC) solver from *lightsim2grid* package is used for data generation and as the baseline for benchmarking. For more reliable results, all the experiments are executed five times and the results are reported using mean and standard deviation.

The hyper-parameters of all the introduced techniques are fine-tuned using the Nevergrad library [17]. Moreover, a dynamic learning rate reducing technique (i.e., reduce learning rate on plateau) is used, which exploits the validation loss for a better learning rate adjustment during the training of the models. The training of physics-informed models often starts with rapid improvements (as the model learns the rough shape of the solution). As it tries to satisfy stricter physics constraints, loss reduction slows down and avoids the overfitting and premature stagnation. The details concerning the hyper-parameters could be seen in the Github repository.

4.2 Benchmark testbed and evaluation criteria

The Learning Industrial Physical Simulation (LIPS) framework [13] is used as the benchmark testbed and evaluation pipeline. It suggests to evaluate the performance of the models with respect to four categories of evaluation criteria:

- *Machine learning related*: The precision of the models to predict the output variable (the phasors θ) is computed using the mean absolute error (MAE) and mean absolute percentage error computed on the 10% of highest values (MAPE90);
- *Physics compliance*: To examine the physical compliance of different approaches, we propose to evaluate them with respect to a set of power grid physical laws considered as metrics. These evaluation criteria and the corresponding metrics are detailed in Table 3;

Table 3: Physical constraints considered for the evaluation of *physics compliance* category. The predictions should be conform, at best, to all of these constraints.

ID Type	Measure	Description
Basic		
P1 Losses positivity	$\frac{1}{L} \sum_{\ell} \mathbb{1}_{(\hat{p}_{ex}^{\ell} + \hat{p}_{or}^{\ell} < 0.)}$	Proportion of negative energy losses
P2 Disconnected Line	$\frac{1}{L_{disc}} \sum_{\ell} \mathbb{1}_{(\hat{x}_{ex}^{\ell} + \hat{x}_{or}^{\ell} > 0.)}$	Proportion of non-null a , p or q values
P3 Energy Losses	$\frac{\sum_{\ell=1}^L (\hat{p}_{ex}^{(\ell)} + \hat{p}_{or}^{(\ell)})}{Gen} \in [0.005, 0.04]$	Energy losses range consistency
Uni-dimension Law		
P4 Global Conservation	$MAPE((Prod - Load) - (\sum_{\ell=1}^L (\hat{p}_{ex}^{\ell} + \hat{p}_{or}^{\ell})))$	Mean energy losses residual
P5 Local Conservation	$MAPE((p_k^{prod} - p_k^{load}) - (\sum_{l \in \text{neig}(k)} \hat{p}_k^{\ell}))$	Mean active power residual at nodes

- *Out-of-distribution generalization*: To evaluate the generalization capability of the models, the same machine learning metrics are computed using the OoD dataset. This dataset includes a slightly different distribution (two simultaneous disconnected power lines) compared to training and test datasets (only one power line disconnection);
- *Industrial readiness*: To evaluate the industrialization capability of the models, we have considered the acceleration or speed-up over the physical solver (DC solver). To consider the scaling capability of different studies, we have also considered two power grids of different sizes, namely IEEE 14 and NeurIPS small including 36 substations (see Table 2).

4.3 Results

Table 4 summarizes the benchmark results for the various studies introduced in the previous section. These studies are compared through the four previously mentioned categories of evaluation criteria. The performances are computed between the predicted phasors ($\hat{\theta}$) and ground-truth values (DC-solver). For the physics compliance category, we only consider the local conservation error in terms of the violation percentage (P_5 in Table 3), as all the methods satisfy the remaining considered physical constraints. The best performances for training-based approaches (excluding the MP Opt) are highlighted in bold.

Table 4: Benchmark table. The results are presented using mean \pm standard deviation (over 5 runs). The compared methods are multi-layer perceptrons (MLP), MLP with physical constraint as regularization (MLP Reg), message-passing and physics constraint as optimization (MP Opt) and physics-informed message-passing with warm initialization (PIMP). The output variables are phasors at each node of the graph (θ). The considered criteria are Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) computed on 10% of lines with highest values (MAPE90), Inference time represented by the speed-up with respect to physic solver (Inf. speed-up), Out-of-distribution Generalisation (OOD Gen.), and physics compliance category (Physics comp.) computing (P5) local conservation laws on both test and OOD datasets.

Methods		Variable	Evaluation Criteria Categories						
			ML-related		Readiness	OOD Gen.		Physics Comp.	
			MAE	MAPE90	Inf. speed-up	MAE	MAPE90	Test	OOD
IEEE 14	MLP	θ	1e-2±1e-3	2e-3±2e-4	4	4e-1±1e-2	1e-1±2e-3	40%±1%	47%±.5%
	MLP Reg	θ	6e-2±2e-2	1e-2±5e-3	4	6e-1±4e-2	1e-1±9e-3	35%±1%	42%±1%
	MP Opt	θ	8e-4 ± 7e-9	4e-3 ± 7e-9	1	1e-2±3e-9	4e-3±8e-9	0%	0%
	PIMP	θ	2e-3±2e-5	7e-4±6e-6	2	2e-2±1e-4	8e-3±5e-5	1%±.1%	13%±.7%
NeurIPS (36 nodes)	MLP	θ	1e-1±2e-3	5e-2±1e-3	8	2e-1±2e-3	1e-1±1e-3	50%± 1%	50%±1%
	MLP Reg	θ	7e-1±2e-1	4e-1±8e-2	8	9e-1±2e-2	5e-1±8e-2	45%±1%	47%±1%
	MP Opt	θ	4e-3±1e-9	2e-3±6e-9	1	1e-2±1e-9	4e-3±4e-9	0%	0%
	PIMP	θ	4e-2±3e-5	2e-2±2e-3	4	6e-2±1e-3	3e-2±4e-3	20%±.5%	26%±.5%

Analysis of IEEE14 environment We can observe that the best results are achieved using the *MP Opt*, which updates the target values based on physics equations and yields a very low standard deviation. It may seem to be the normal behavior, as this approach does not include any stochastic initialization or operation during the optimization. It requires 100 iterations (message-passing layers) to converge to the optimal solution, where the local conservation error is very close to zero. One of the objectives of considering this study was to analyze whether the addition of some

learnable parameters for the initialization phasors $\theta^{(0)}$ may allow reducing the number of required message-passing layers.

To conduct this experimentation, we have reduced the number of message-passing layers by half (60 layers) and added an MLP with four hidden layers for the initialization of phasors (*PIMP*). It shows significantly better performances than MLP-based approaches and is comparable to that of the pure optimization-based approach. The main difference could be seen with respect to the physics compliance criteria. Despite the notable improvement obtained over MLP-based approaches (reducing the local conservation error by more than 30%), the compliance remains lower than *MP Opt* for the OoD dataset (13% of law violation). Adding more message-passing layers allows to improve further the physics compliance, which indeed is directly related to the model complexity.

The two MLP-based approaches obtain very close performances with respect to ML-related criteria for both test and OoD datasets. They show also the best speed-ups behavior with respect to the DC physical solver. The regularization through the penalization of the physics term allows to reduce slightly the local conservation error for *MLP Reg*. During the experimentation, we have observed that this approach requires more iterations (epochs) for the better convergence than the MLP without any regularization. It should also be mentioned that the DC solver implemented in *lightsim2grid* is very fast and the optimization could be computed in parallel for the considered scenario (a scenario known also as security analysis). This makes acceleration improvement quite challenging.

Analysis of NeurIPS environment (36 nodes) A similar pattern could also be observed for the larger grid including 36 substations. For this more complex power grid, the *MP Opt* method requires substantially more iterations to converge (350 iterations or message-passing layers). The performance slightly degrades compared to the smaller grid across all evaluation criteria, with the exception of the speed-up. This highlights the role that hybrid models can play when applied to more realistic, large-scale grids. For the sake of this experimentation, the number of message passing layers (300 message passing layers) required by *MP Opt* to fully satisfy the physical constraints is halved for *PIMP*. We can observe clearly much better compliance with local conservation laws compared to MLP-based approaches. At the same time, an increasing trend in physical constraint violation could be observed which is correlated to network complexity, and may necessitate a more rigorous enforcement of such physical constraints.

Convergence analysis The convergence behavior of these approaches are compared in Figure 6 for IEEE14 environment. The convergence is represented using mean losses (solid lines), and the highlighted area around the mean represents the standard deviation over 5 runs. For the approaches requiring training, the validation loss is also shown in green in addition to the training loss which is shown in blue. It can be observed that all the approaches converge to a local optimum with slightly different convergence behaviors. The MLP-based approach (see Figure 6a) which does not consider the physical constraint converges enough fast with low variation of the loss over the data. However, the regularized MLP which considers the physical constraint

as penalization of the loss, shows some variations in the beginning of the training due to the high values of local conservation error, and stabilizes when the predictions approach the ground-truth values (see Figure 6b). The *MP Opt* approach does not requiring training and optimizes the local conservation error directly through the message passing layers and updating the phasors. Finally, the *PIMP* with warm initialization (see Figure 6d) presents more stabilized training curve than those obtained by MLP, and converges faster than the MP optimization-based approach.

To analyze the difference between two Message-passing based approaches in greater details, Figure 7 compares their convergence using the same graphic (with both natural and log scale of local conservation errors on y-axis). For consistency, we have considered the same number of iterations for this analysis. As can be seen, the initialization of phasors using MLP enables the faster convergence of the approach (*PIMP*), despite the reduced number of message-passing layers.

Finally, to better understand the impact of data and physics terms during the training of *MLP Reg* and *PIMP* initialized using MLP, Figure 8 demonstrates their behavior in separate graphics. As can be seen in Figure 8a, the physics loss (orange curve) for *MLP Reg* shows high variability at the beginning of the training which

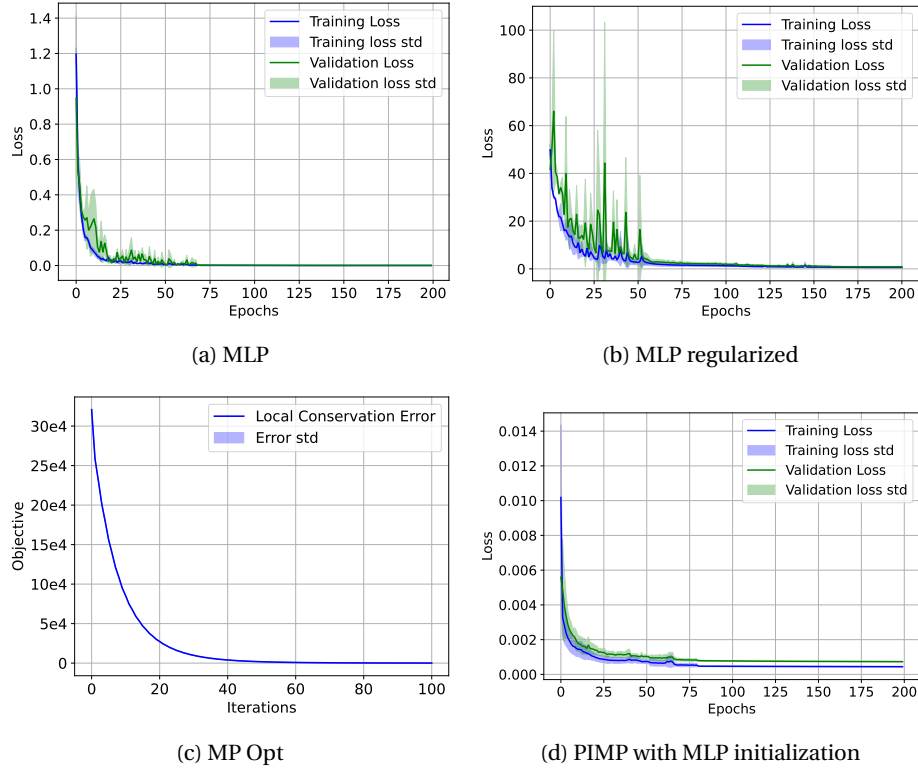


Fig. 6: Convergence behavior of different approaches for IEEE 14 environment

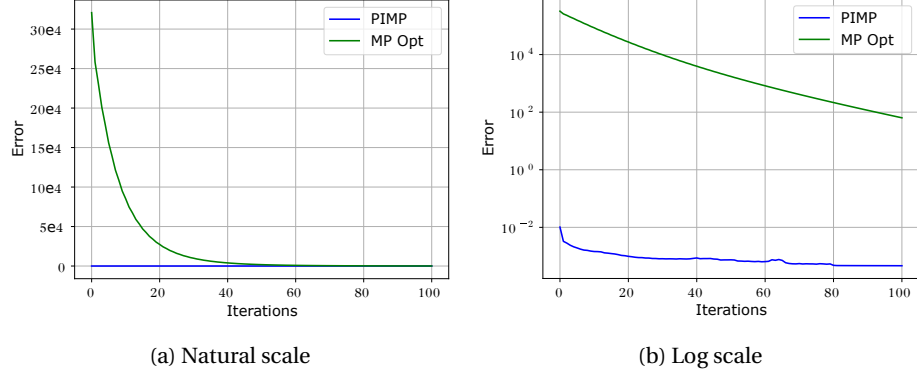


Fig. 7: Comparison of the training curve convergence between Message-Passing with optimization (*MP Opt*) and Message-Passing with MLP initialization (*PIMP*).

stabilizes over time. However, it never reaches the optimal solution. On the other hand, *PIMP* shows more stabilized physics loss behavior from the beginning of the training and overlaps the data loss early. It could be explained by the fact that the message-passing based approaches takes advantage of the data structure (grid topology) and allows to be consistent to physical context by design.

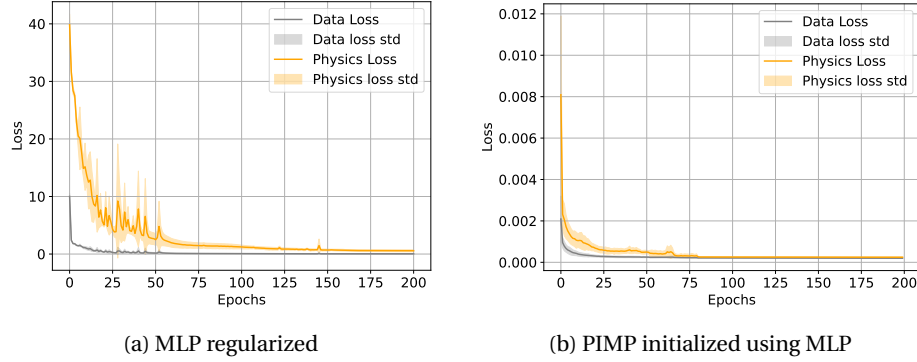


Fig. 8: Convergence behavior of physics loss vs. data loss

5 Conclusion and perspectives

This paper proposed a study design to benchmark and investigate the inner workings of physics-informed neural networks. A unified benchmarking framework was employed to enable the comprehensive evaluation of various methods. The experimental results indicate that pure machine learning-based techniques can achieve high accuracy; however, their performance often fall short in meeting real-world requirements, particularly in terms of physics compliance and out-of-distribution generalization. In

contrast, approaches that incorporate physics knowledge – either as a regularization term in the loss function or as part of the optimization objective – demonstrated superior performance across most evaluation criteria.

Furthermore, the experiments revealed that physics-informed approaches exhibit strong potential in low-data regimes. These methods outperformed classical approaches when the number of training samples was reduced. All studies presented in this paper are fully reproducible, with implementations available in the GitHub repository, and are evaluated using a standardized evaluation pipeline, thereby supporting further research in this domain.

For didactic purposes, this study focused on modeling power flow using a DC solver. As a direction for future work, we plan to extend this framework to the more complex AC power flow, which additionally requires balancing voltage magnitudes. In the current work, equal importance was assigned to the data and physics components of the loss function. However, several techniques have been proposed in the literature to improve convergence and enhance physics compliance. For instance, gradually increasing the weight of the physics-based loss term has been shown to be effective [7, 10].

The application on real-world data may often encounter inconsistencies or other data issues. We plan to mitigate them by incorporating the mechanisms for handling missing or corrupted data through, for example, robust loss functions or anomaly detection, improving the solver stability and results reliability. We also observed that increasing grid complexity leads message-passing-based methods to require more iterations (or layers) to converge. Consequently, their computational complexity may increase linearly with the grid complexity. Future studies will focus on incorporating learnable parameters into the message-passing mechanism, moving toward the use of graph neural networks. This approach is expected to reduce the number of required iterations.

Acknowledgments

The research leading to this work is part of the AI4REALNET (*AI for REAL-world NETwork operation*) project, which received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101119527, and from the Swiss State Secretariat for Education, Research and Innovation (SERI). This project is funded by the European Union and SERI. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union and SERI. Neither the European Union nor the granting authority can be held responsible for them.

References

1. Babatunde, O.M., Munda, J.L., Hamam, Y.: Power system flexibility: A review. *Energy Reports* **6**, 101–106 (2020)
2. Baker, K.: Learning warm-start points for ac optimal power flow. In: 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6. IEEE (2019)

3. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al.: Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* (2018)
4. Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., Zdeborová, L.: Machine learning and the physical sciences. *Reviews of Modern Physics* **91**(4), 045002 (2019)
5. Donnot, B., Donon, B., Guyon, I., Liu, Z., Marot, A., Panciatici, P., Schoenauer, M.: Leap nets for power grid perturbations. *arXiv preprint arXiv:1908.08314* (2019)
6. Donon, B., Clément, R., Donnot, B., Marot, A., Guyon, I., Schoenauer, M.: Neural networks for power flow: Graph neural solver. *Electric Power Systems Research* **189**, 106547 (2020)
7. Hanna, J.M., Vignon-Clementel, I.E., Talbot, H.: Improved physics-informed neural networks loss function regularization with a variance-based term. Available at SSRN 5269341 (2024)
8. Hasan, F., Kargarian, A., Mohammadi, A.: A survey on applications of machine learning for optimal power flow. In: 2020 IEEE Texas Power and Energy Conference (TPEC). pp. 1–6. IEEE (2020)
9. Kasim, M., Watson-Parris, D., Deaconu, L., Oliver, S., Hatfield, P., Froula, D., Gregori, G., Jarvis, M., Khatiwala, S., Korenaga, J., et al.: Building high accuracy emulators for scientific simulations with deep neural architecture search. *Machine Learning: Science and Technology* **3**(1), 015013 (2021)
10. Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., Mahoney, M.W.: Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems* **34**, 26548–26560 (2021)
11. Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., et al.: Learning skillful medium-range global weather forecasting. *Science* **382**(6677), 1416–1421 (2023)
12. Lei, X., Yang, Z., Yu, J., Zhao, J., Gao, Q., Yu, H.: Data-driven optimal power flow: A physics-informed machine learning approach. *IEEE Transactions on Power Systems* **36**(1), 346–354 (2020)
13. LEYLI ABADI, M., Marot, A., Picault, J., Danan, D., Yagoubi, M., Donnot, B., Attoui, S., Dimitrov, P., Farjallah, A., Etienam, C.: Lips-learning industrial physical simulation benchmark suite. *Advances in Neural Information Processing Systems* **35**, 28095–28109 (2022)
14. Lin, N., Orfanoudakis, S., Cardenas, N.O., Giraldo, J.S., Vergara, P.P.: Powerflownet: Power flow approximation using message passing graph neural networks. *International Journal of Electrical Power & Energy Systems* **160**, 110112 (2024)
15. Lopez-Garcia, T.B., Domínguez-Navarro, J.A.: Power flow analysis via typed graph neural networks. *Engineering Applications of Artificial Intelligence* **117**, 105567 (2023)
16. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561* (2017)
17. Rapin, J., Teytaud, O.: Nevergrad - A gradient-free optimization platform. <https://GitHub.com/FacebookResearch/Nevergrad> (2018)
18. Saadat, H., et al.: Power system analysis, vol. 2. McGraw-hill (1999)
19. Stott, B.: Review of load-flow calculation methods. *Proceedings of the IEEE* **62**(7), 916–929 (1974)
20. Tompson, J., Schlachter, K., Sprechmann, P., Perlin, K.: Accelerating eulerian fluid simulation with convolutional networks. *ArXiv: 1607.03597* (2016)
21. Vlach, J., Singhal, K.: Computer methods for circuit analysis and design. Springer Science & Business Media (1983)
22. Xavier, Á.S., Qiu, F., Ahmed, S.: Learning to solve large-scale security-constrained unit commitment problems. *INFORMS Journal on Computing* **33**(2), 739–756 (2021)