Boosting Reinforcement Learning for Tertiary Voltage Control by Transfer Learning from a Supervised Power Flow Simulation Task

François Cubélier
Dept of EE & CS - Montefiore Institute
Université de Liège
Liège, Belgium
f.cubelier@uliege.be

Balthazar Donon
R&D Department
RTE (Réseau de Transport d'Électricité)
Paris, France
balthazar.donon@rte-france.com

Louis Wehenkel
Dept of EE & CS - Montefiore Institute
Université de Liège
Liège, Belgium
l.wehenkel@uliege.be

***Abstract*—This work explores how transfer learning can improve reinforcement learning for tertiary voltage control, which is a simulation-intensive process. The model is pre-trained over the supervised learning of a power flow simulator, as a way of incorporating physics inside the model. Two transfer strategies are proposed and compared against a transfer-free baseline. The *case60nordic* test case, which provides diversified operating conditions, including topological variations, is used to assess performance. Results indicate that fine-tuning the pre-trained model can effectively improve performance or reduce training time on the target task. This study emphasizes the potential of transfer learning for accelerating the training of power grid control downstream tasks.**

***Index Terms*—Transfer learning, Power systems, Tertiary voltage control, Graph neural networks**

I. Introduction

Tertiary Voltage Control (TVC) is crucial for the stability and efficiency of transmission grid operation. Inadequate TVC may lead to grid instability, equipment damage and system collapse in the worst-case scenario. The growing integration of intermittent renewable energy, notably wind and solar, makes voltage control even more critical to operate the grid properly. Deep learning, and particularly Graph Neural Networks (GNNs), which are designed to handle graph data, are very promising for transmission grid challenges. Examples of GNNs applied on transmission grids include grid control [1]–[3], forecasting [4], [5] and state estimation [6], [7].

This article builds upon the GNN-based Reinforcement Learning (RL) approach to TVC proposed in [8]. A very large number of power flow simulations is required by this method to assess actions tried out during the learning iterations, which slows down the overall learning process. Available machine learning techniques could attenuate this downside. In particular, this article focuses on transfer learning [9]–[11], a machine learning methodology that aims at reusing knowledge acquired by a model learnt to solve a source task to improve the learning of solutions for other target tasks. Transfer learning methods have already been applied successfully in power systems to reduce training time or accommodate a lack of labeled data in downstream target tasks [12], [13].

More specifically, to solve the TVC task by RL we investigate the use of power flow (PF) simulation [14]–[18] as a source task, as it captures meaningful physical information about the grid that the TVC task could reuse. Whereas the RL-based TVC task needs on-the-fly simulations during training, the supervised PF simulation task only requires computing the ground truths once before training using the chosen power flow simulator. In practice, the considered RL-based TVC training involved over 10^8 power flow simulations, contrasting to the 10^5 required for pre-training a machine learning approximation of the PF simulator (on the same dataset).

In this work, we assess the potential of supervised pre-training on the PF simulation task to improve the RL-based TVC training from our previous work [8]. Our results on the *case60nordic* test case [8], [19] show that fine-tuning models stemming from our proposed pre-training outperforms training from scratch, yielding better objective values and fewer voltage violations while accelerating training speed.

The paper is organized as follows. Section 2 details the methodology, explaining both the source task of PF simulation and the target task of TVC, as well as the transfer strategies. In Section 3, the experimental settings and results are presented with a comparison of two different transfer learning strategies against training from scratch and a classical optimization baseline [8]. Finally, Section 4 concludes the paper with key insights from the study and openings for potential future research directions.

II. Methodology

This section outlines the proposed transfer learning methodology. The process is designed to exploit knowledge gained from the source task, PF simulation, to enhance performance on the target task, TVC. The approach is decomposed into four key components: (A) a robust data representation and specialized GNN architecture designed for power systems, (B) PF simulation task definition and training methodology, (C)

979-8-3315-2503-3/25/$31.00 ©2025 IEEE

TVC task definition and training procedure, and (D) transfer learning from PF simulation to TVC. Figure 1 provides a visual overview of our proposed pre-training and transfer stages.

A. Data representation and graph neural network

We employ the Hyper-Heterogeneous Multi Graph (H2MG) [20] framework to represent power systems data. It renders the diversity of object classes encountered in the representation of power systems. It allows to represent these objects as they are without simplifying them into standard node-and-edge graphs.

A power system operating condition, denoted by x, contains numerical features as well as topological features (i.e., connectivity between components). The numerical features of x include, among others:

- Active and reactive powers of all loads,
- Voltage bounds of all buses,
- Active powers, reactive power bounds, and voltage setpoints of all generators,
- Admittances of all lines and all transformers,
- Nominal reactive powers of all shunts.

In the H2MG framework, an operating condition x is composed of multiple classes of objects (generators, loads, lines, etc.). Each element of a class is described by its numerical features and its connection to other objects.

We use the companion H2MGNODE architecture to handle this data. It consists of a Graph Neural Network (GNN) designed for power systems that leverages Neural Ordinary Differential Equations (NODE) [21]. This GNN architecture is used to map the input x to a task-specific output $y \in \mathcal{Y}(x)$,

$$\hat{y}_\theta(x) = \mathbf{D}_{\theta_D} \circ \mathbf{C}_{\theta_C} \circ \mathbf{E}_{\theta_E}(x). \tag{1}$$

This architecture consists of three successive components: the encoding $\mathbf{E}$, the coupling $\mathbf{C}$, and the decoding $\mathbf{D}$, with corresponding parameters θ_E, θ_C and θ_D. During encoding, every object feature vector is mapped into an encoded version, denoted $\tilde{x} := E_{\theta_D}(x)$, by a Multi Layer Perceptron (MLP) specific to the class of the object. The coupling function, central to the GNN, leverages message passing to generate latent variables at each bus location in the power grid. The latent variables $h := C_{\theta_C}(\tilde{x})$ are progressively computed during the resolution of the NODE by combining information from the neighboring object's encoded features via MLPs. Finally, class-specific MLP decoders combine these latent variables with the encoded input features to generate the desired output features $\hat{y} := \mathbf{D}_{\theta_D}(h, \tilde{x})$. We refer the interested reader to [8], [20] for a more precise description and motivation of the H2MGNODE model.

B. Source task: Power flow simulation

The source task consists in predicting the power flow simulation outputs, denoted y^{pf}, knowing its input features x. It is framed as a supervised learning problem. Here we consider, without limitation, a static AC power flow simulation task based on the Newton-Raphson method while ensuring that reactive generator limits are respected. This source task thus incorporates relevant knowledge about the physics of power grids.

The considered power flow simulation outputs y^{pf} contain the following features:

- Voltage magnitudes and phase angles at all buses,
- Active and reactive powers of all generators,
- Active and reactive power flows of all branches (*i.e.* lines and transformers),
- Current magnitudes and loading percentages of all branches.

The model is trained on this task by minimizing the mean squared error (MSE) between the model's predictions and the ground truths precomputed by the Newton-Raphson method. The overall loss function of the PF simulation problem is:

$$\mathcal{L}^{pf}(\theta^{pf}) = \mathbb{E}_{x \sim p(\cdot)} \left[MSE\left(\hat{y}^{pf}_{\theta^{pf}}(x), y^{pf}(x) \right) \right], \tag{2}$$

where p is the considered distribution of operating conditions.

C. Target task: Tertiary voltage control

The target task consists in controlling generator voltage setpoints to maintain voltages within operational limits, as defined in [8]. The task amounts to minimizing a cost function $c(x, y^{vc})$, which takes as inputs the operating condition x and the generator voltage setpoints y^{vc}, and balances three objectives: minimizing voltage and current operational limit violations and reducing Joule losses in the grid,

$$y^{vc}(x) \in \underset{y^{vc} \in \mathcal{Y}^{vc}(x)}{\arg\min} \; c(x, y^{vc}). \tag{3}$$

To solve this problem for every x, we teach a H2MGNODE to map the input features x to a solution $\hat{y}^{vc}$ via RL. Following [8], a stochastic policy $\Pi_{\theta^{vc}}$ for voltage setpoints is defined as a multivariate Gaussian distribution, with its mean predicted by the neural network $\hat{y}^{vc}_{\theta^{vc}}$: $\Pi_{\theta^{vc}}(\cdot|x) = \mathcal{N}(\hat{y}^{vc}_{\theta^{vc}}(x), \sigma^2 \mathbb{I})$, where $\sigma > 0$ is a fixed parameter and $\mathbb{I}$ is the identity matrix.

The aim is to minimize the expected cost under this stochastic policy:

$$\mathcal{L}^{vc}(\theta^{vc}) = \mathbb{E}_{\substack{x \sim p(.) \\ y^{vc} \sim \Pi_{\theta^{vc}}(.|x)}} [c(x, y^{vc})]. \tag{4}$$

The parameters θ^{vc} are trained to minimize this loss by following the REINFORCE method. According to the well-known “log-trick”, the gradient of $\mathcal{L}^{vc}$ can be formulated as follows:

$$\nabla_{\theta^{vc}} \mathcal{L}^{vc}(\theta^{vc}) = \mathbb{E}_{\substack{x \sim p(.) \\ y^{vc} \sim \Pi_{\theta^{vc}}(.|x)}} [c(x, y^{vc}) \nabla_{\theta^{vc}} \ln \Pi^{vc}_{\theta^{vc}}(y^{vc}|x)]. \tag{5}$$

This gradient is estimated via Monte Carlo simulation over mini-batches of operating conditions x, with power flow simulations (same simulator as above) used to evaluate the cost function for voltage control candidates (see [8] for the exact algorithm statement).

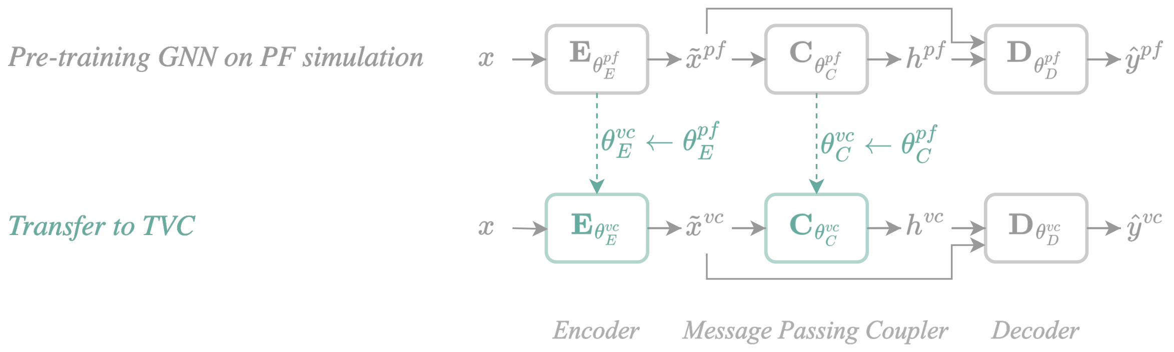
Pre-training GNN on PF simulation
x
$\mathbf{E}_{\theta_E^{pf}}$
$\tilde{x}^{pf}$
$\mathbf{C}_{\theta_C^{pf}}$
h^{pf}
$\mathbf{D}_{\theta_D^{pf}}$
$\hat{y}^{pf}$
$\theta_E^{vc} \leftarrow \theta_E^{pf}$
$\theta_C^{vc} \leftarrow \theta_C^{pf}$
Transfer to TVC
x
$\mathbf{E}_{\theta_E^{vc}}$
$\tilde{x}^{vc}$
$\mathbf{C}_{\theta_C^{vc}}$
h^{vc}
$\mathbf{D}_{\theta_D^{vc}}$
$\hat{y}^{vc}$
Encoder
Message Passing Coupler
Decoder

Fig. 1: GNN-based transfer learning workflow for TVC of transmission systems. The process begins by pre-training the GNN on the source task, PF simulation, through supervised learning. The pre-trained model is then transferred to the target task, TVC, and adapted using RL, with two strategies: *PF Frozen* or *PF Fine-Tuned*. Transfer Learning relies on the fact that both tasks share the same input representation x and the same parametric architecture for their encoder and coupler components.

D. Transfer learning

The Transfer Learning procedure begins with the pre-training of the parameters θ^{pf} on the source task, PF simulation, as described earlier (II-B). Then, the trained encoder and coupler weights, θ_E^{pf} and θ_C^{pf}, are injected into the TVC model, while its decoder weights θ_D^{vc} are randomly initialized. Notice that this is only possible if both models share the same encoder and coupler architectures.

The model is then trained on the target task (II-C) following one of the two proposed transfer strategies: *PF Frozen* or *PF Fine-Tuned*. In both strategies, the pre-trained parameters serve as initialization of the encoder-coupler.

1) PF Frozen: The pre-trained encoder-coupler parameters remain fixed while only the TVC decoder is trained. With this approach, latent variables can be pre-computed for each operating condition, offering a computational advantage during training. Notice that decoding does not involve message passing. Therefore, this strategy assumes that pre-trained latent variables contain sufficient information to determine near-optimal generator voltage setpoints.

2) PF Fine-Tuned: Both the pre-trained encoder-coupler parameters and the TVC decoder are trained together on the target task.

III. Experiments

This section details the settings of the experiments and their results on the *case60nordic* test case. Results on the source task summarise the pre-trained model performance. For the target task TVC, the cost evolutions during training are analyzed, as well as the resulting operational limit violations obtained by the different models, showing the efficacy of fine-tuning.

A. Settings

Let us first describe the experimental settings. Each experiment is run 5 times with different random seeds.

Case study: Experiments are run on the *Standard* dataset [1] from [8] (100,000 operating conditions for training, 2,000 for validation, and 10,000 for testing), generated from the *case60nordic* power grid [19]. This dataset, used in previous work, is representative of the variability of real-life operating conditions. It displays topological variations with up to four lines disconnected, randomization of the loads (both total and individual demands) and generation (from 6 to 19 generators). The dataset generation process is described in [22]. In our study, generator voltage magnitude setpoints (initially fixed at 1 p.u. in [8]) are uniformly and independently sampled between 0.9 and 1.1 p.u. This sampling may lead to power flow non-convergence (in practice, less than 1% of the time), in which case resampling is iteratively applied until convergence. Power flow simulations are run with the PandaPower library [23]. Input features are normalized following the approach described in [22].

ACOPF solver: A classical AC Optimal Power Flow (ACOPF) solver serves as the golden standard, as described in [22]. Notice that this solver is known to adapt poorly to large power grids and discrete control variables, thus making our learning-based methodology competitive.

Baseline: The *Baseline* for TVC follows the methodology introduced in our previous work [8] (see also II-C). For simplicity and excluding negligible changes, we reuse the same hyper-parameters as in the original article. For the neural network architecture, we use a latent dimension of 64, two hidden layers of hidden size 128 and 64 for the MLP encoders and decoders and 1 hidden layer of size 128 for the coupling MLPs. The model parameters θ^{vc} are optimized by mini-batch gradient descent on the loss (4) for 200,000 iterations with the Adam optimizer [24] (with standard parameters). The learning rate is 3×10^{-4}, and the batch size is 32. The average validation cost is monitored at every epoch, and only the model with the lowest cost is retained, as a form of early stopping.

Transfer: For the pre-training stage, we learn the parameters θ^{pf} by minimizing the loss function (2) for 200,000 iterations. We use the same hyper-parameter choices for the neural architecture and the Adam optimizer as for the *Baseline*.

The output features y^{pf} are center-reduced using mean and standard deviation over the training set. Model selection is

[1] Dataset available at https://zenodo.org/records/10825468

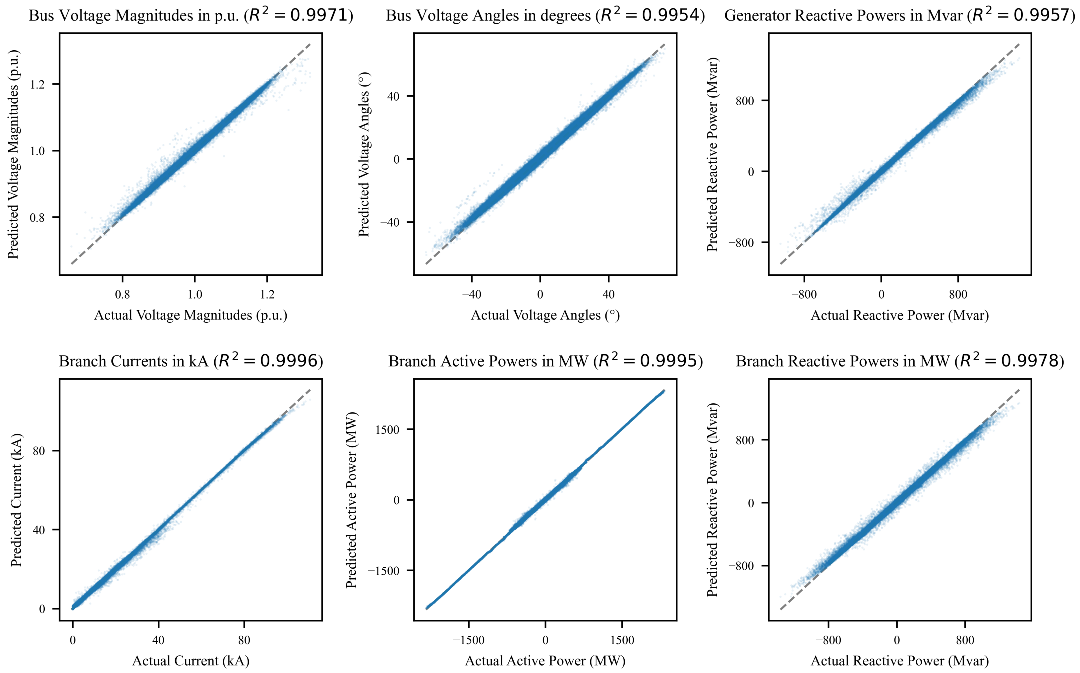

Fig. 2: Power flow simulation pre-taining: Scatter plots comparing actual values and predicted values with R^2 scores on the test set for various output features.

based on the R^2 metric, a standard regression measure, to assess whether a model optimized for PF simulation can effectively capture features relevant to the TVC task. After every training epoch, the feature-averaged R^2 score is computed over the validation set and only the best performing model is kept.

For the transfer strategies *Proxy Frozen* and *Proxy Fine-Tuned*, we employ the same hyper-parameters as the *Baseline* for training.

B. *Results*

Though PF simulation is not the end goal of this study, we showcase the results of the pre-trained model on the source task to get a better picture of the overall transfer process. We then proceed with the TVC results.

1) PF simulation task: As shown in Figure 2, the pre-trained model demonstrates R^2 scores over 0.995 for the different categories of features on the source task of PF simulation, showing how effectively it captured the underlying physics of power grid behavior. The scatter plots show predicted and ground truth values for each output feature. The points remain close to the diagonal line, which illustrates the model's accuracy.

2) TVC task: To assess the effectiveness of transfer learning, three distinct training strategies were evaluated: the *Baseline* was compared against two proposed transfer learning strategies: *PF Frozen* and *PF Fine-Tuned*.

a) Cost Evolution: Figure 3 illustrates the cost evolution on the validation set during training for each strategy. The *PF Fine-Tuned* strategy achieved the best final cost among all the methods. It reached the *Baseline* final cost in roughly two-thirds of the training steps, showing the advantage of smart initialization to accelerate training. The *PF Frozen*, however, ended up with the worst final cost. This could be due to the inability of the frozen encoder and coupler to capture the information required by the decoder to fully adapt to the TVC task.

b) Constraint Violations: Table I outlines the resulting operational constraint violations for each strategy. Violations encompass both bus voltage violations (*i.e.* outside of the [0.9, 1.1] p.u. range) and branch thermal limit violations. We observe that *PF Fine-Tuned* obtained the least violations, only 3.8% against 4.6% for the *Baseline*, closing the gap with the ACOPF solver (1.0%). On the other hand, the *PF Frozen* failed to adapt to TVC, with over 24% situations violating operational limits. This experiment outlines the advantage of fine-tuning in terms of convergence speed and operational constraint handling.

TABLE I: Mean percentage of operating conditions with constraint violations ± standard error for three learning strategies on the test set, with ACOPF reference.

Operating conditions with violation.	
Baseline (no transfer)	$4.6 \pm 0.1\%$
PF Frozen	$24.1 \pm 0.7\%$
PF Fine-Tuned	$\mathbf{3.8} \pm 0.1\%$
ACOPF solver	$1.0 \pm 0.0\%$

IV. Conclusion

This study highlights the potential of transfer learning as a relevant tool for TVC of electric power systems, using PF simulation as the source task. Two transfer strategies were

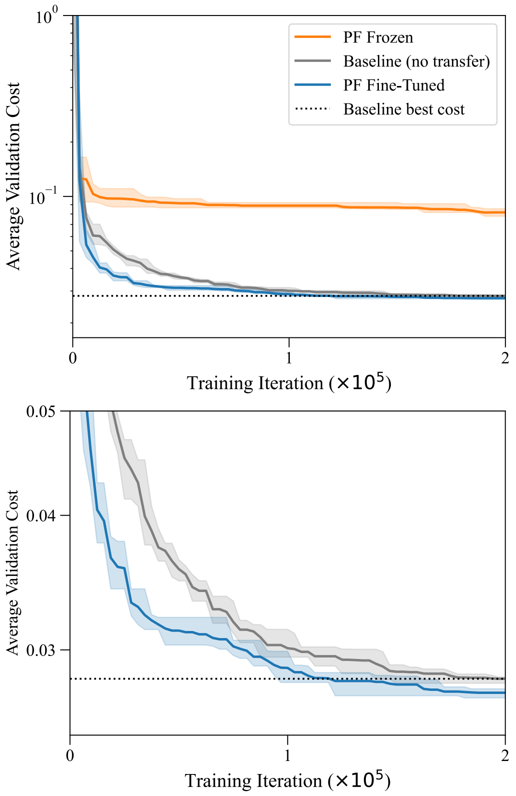

Fig. 3: Evolution of the best average cost reached on the validation set during TVC training for three different strategies. The line represents the mean over the 5 runs, and the shaded area displays the minimum and maximum costs over the 5 runs. The *PF Fine-Tuned* training curve (blue) crosses the *Baseline* best cost (dotted line) in two-thirds the training steps, illustrating its faster convergence.

proposed and compared against the previous RL-based TVC baseline. The first, *PF Fine-Tuned*, where pre-trained weights are fine-tuned, improved over the baseline, reaching the same performance in two-thirds of the training steps. Furthermore, given the same training budget, it surpassed the baseline performance in terms of operational constraint violation. The second, *PF Frozen*, where pre-trained weights are frozen, failed to achieve satisfactory performances.

Future research could investigate alternative source tasks and more advanced transfer learning strategies to further enhance performance. Additionally, this study could be extended to explore other optimization and control tasks of electric power systems. Furthermore, future work could include the application of this approach to larger test cases.

Acknowledgments

The authors are grateful to the operators of the Alan GPU cluster at the University of Liège, where the experiments for this study were conducted.

References

[1] M. Hassouna, C. Holzhüter, P. Lytaev, J. Thomas, B. Sick, and C. Scholz, "Graph reinforcement learning for power grids: A comprehensive survey," *Preprint arXiv:2407.04522*, 2024.

[2] S. Liu, C. Wu, and H. Zhu, "Topology-aware graph neural networks for learning feasible and adaptive ac-opf solutions," *IEEE transactions on power systems*, vol. 38, no. 6, pp. 1–11, 2023.

[3] D. Owerko, F. Gama, and A. Ribeiro, "Optimal power flow using graph neural networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5930–5934, 2020.

[4] C. Wei, D. Pi, M. Ping, and H. Zhang, "Short-term load forecasting using spatial-temporal embedding graph neural network," *Electric power systems research*, vol. 225, pp. 109873–, 2023.

[5] D. Beinert, C. Holzhüter, J. M. Thomas, and S. Vogt, "Power flow forecasts at transmission grid nodes using graph neural networks," *Energy and AI*, vol. 14, p. 100262, 2023.

[6] Q.-H. Ngo, B. L. Nguyen, T. V. Vu, J. Zhang, and T. Ngo, "Physics-informed graphical neural network for power system state estimation," *Applied Energy*, vol. 358, p. 122602, 2024.

[7] O. Kundacina, M. Cosovic, and D. Vukobratovic, "State estimation in electric power systems leveraging graph neural networks," *Preprint arXiv:2201.04056*, 2022.

[8] B. Donon, F. Cubélier, E. Karangelos, L. Wehenkel, L. Crochepierre, C. Pache, L. Saludjian, and P. Panciatici, "Topology-aware reinforcement learning for tertiary voltage control," *Electric Power Systems Research*, vol. 234, p. 110658, 2024.

[9] J. West, D. Ventura, and S. Warnick, "Spring research presentation: A theoretical foundation for inductive transfer," *Brigham Young University, College of Physical and Mathematical Sciences*, vol. 1, no. 08, 2007.

[10] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[11] M. Gholizade, H. Soltanizadeh, M. Rahmanimanesh, and S. S. Sana, "A review of recent advances and strategies in transfer learning," *International Journal of System Assurance Engineering and Management*, vol. 16, pp. 1123–1162, Mar. 2025.

[12] R. Al-Hajj, A. Assi, B. Neji, R. Ghandour, and Z. Al Barakeh, "Transfer learning for renewable energy systems: A survey," *Sustainability*, vol. 15, no. 11, 2023.

[13] F. M. Shakiba, M. Shojaee, S. M. Azizi, and M. Zhou, "Generalized fault diagnosis method of transmission lines using transfer learning technique," *Neurocomputing*, vol. 500, pp. 556–566, 2022.

[14] 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, 2020.

[15] Z. Kaseb, S. Orfanoudakis, P. P. Vergara, and P. Palensky, "Adaptive informed deep neural networks for power flow analysis," *Preprint arXiv:2412.02659*, 2024.

[16] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE transactions on power systems*, vol. 36, no. 3, pp. 2082–2092, 2021.

[17] G. Parodi, L. Oneto, G. Ferro, S. Zampini, M. Robba, D. Anguita, and A. Coraddu, "Physics informed data driven techniques for power flow analysis," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 33–40, IEEE, 2023.

[18] H.-F. Zhang, X.-L. Lu, X. Ding, and X.-M. Zhang, "Physics-informed line graph neural network for power flow calculation," *Chaos (Woodbury, N.Y.)*, vol. 34, no. 11, 2024.

[19] F. Capitanescu, "Suppressing ineffective control actions in optimal power flow problems," *IET Generation, Transmission & Distribution*, vol. 14, pp. 2520–2527, 2020.

[20] B. Donon, *Deep statistical solvers & power systems applications*. PhD thesis, Université Paris-Saclay, 2022.

[21] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural Ordinary Differential Equations," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[22] B. Donon, F. Cubélier, E. Karangelos, L. Wehenkel, L. Crochepierre, C. Pache, L. Saludjian, and P. Panciatici, "Topology-Aware Reinforcement Learning for Tertiary Voltage Control - Supplementary Material," tech. rep., University of Liège, 2023. https://hdl.handle.net/2268/306778.

[23] L. Thurner, A. Scheidler, F. Schafer, J. H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun, "pandapower - an Open Source Python Tool for Convenient Modeling, Analysis and Optimization of Electric Power Systems," *IEEE Transactions on Power Systems*, 2018.

[24] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations (ICLR)*, 2015.