Power Engineering Letters

# Physics-Aware Neural Dynamic Equivalence of Power Systems

Qing Shen, *Graduate Student Member, IEEE*, Yifan Zhou, *Member, IEEE*, Qiang Zhang, *Senior Member, IEEE*, Slava Maslennikov, *Senior Member, IEEE*, Xiaochuan Luo, and Peng Zhang, *Senior Member, IEEE*

***Abstract*—This letter devises Neural Dynamic Equivalence (NeuDyE), which explores physics-aware machine learning and neural-ordinary-differential-equations (ODE-Net) to discover a dynamic equivalence of external power grids while preserving its dynamic behaviors after disturbances. The contributions are threefold: 1) an ODE-Net-enabled NeuDyE formulation to enable a continuous-time, data-driven dynamic equivalence of power systems; 2) a physics-informed NeuDyE learning method (PI-NeuDyE) to actively control the closed-loop accuracy of NeuDyE without an additional verification module; 3) a physics-guided NeuDyE (PG-NeuDyE) to enhance the method's applicability even in the absence of analytical physics models. Extensive case studies in the NPCC system validate the efficacy of NeuDyE, and, in particular, its capability under various contingencies.**

***Index Terms*—Dynamic equivalence, ODE-Net, physics-informed machine learning, model order reduction.**

## I. Introduction

Discovering reliable dynamic equivalent models of unidentified subsystems or external systems is of critical significance for the resilient operations of large-scale interconnected power systems [1]. However, it is a long-standing obstacle due to the strongly nonlinear dynamics of power systems [2], complicated coherency characteristics [3], unavailable component models, etc. Recently, the wide adoption of PMUs and the high-rate measurement streams generated from them stimulate the development of data-driven dynamic equivalence. While different attempts have been reported, two major challenges remain: I) How to retain the continuous-time dynamic behaviors of dynamic equivalence using discrete-time measurements? II) How to theoretically guarantee the closed-loop performance of dynamic equivalence to support its co-simulation in the entire interconnected power systems?

To bridge the gap, this letter devises Neural Dynamic Equivalence (NeuDyE). The key innovation is the integration of an ODE-Net-enabled dynamic equivalence model and a physics-aware NeuDyE learning to discover a *continuous-time* dynamic equivalence of power grids with explicitly guaranteed *closed-loop dynamic behaviors* under disturbances.

## II. NeuDyE Formulation Via ODE-Net

Denote the subsystem to be equivalenced as the external system (*ExSys*) and the rest as the internal system (*InSys*). Considering the continuous-time dynamic natures of power grids, we formulate the dynamic equivalence of *ExSys* by a set of neural ordinary differential equations (ODE-Net [4]):

$$\dot{x}_{ex} = \mathcal{N}_\theta(x_{ex}, s_{in}) \tag{1}$$

where $x_{ex}$ denotes the selected *ExSys*'s dynamic states; $s_{in}$ denotes *InSys*'s impact on *ExSys*; $\mathcal{N}_\theta$ is an ODE-Net parameterized by $\theta$. Consequently, the entire power grid integrating *ExSys* and *InSys* appears as a physics-neural hybrid system:

$$\begin{cases} \dot{x}_{in} = \mathcal{P}(x_{in}, y_{in}, y_b) & \text{(2a)} \\ \dot{x}_{ex} = \mathcal{N}_\theta(x_{ex}, s_{in}) & \text{(2b)} \\ \mathcal{G}(x_{in}, x_{ex}, y_{in}, y_b) = 0 & \text{(2c)} \end{cases}$$

where $x_{in}$ and $y_{in}$ denote the dynamic and algebraic states of *InSys*; $y_b$ denotes the boundary states between *InSys* and *ExSys*; $\mathcal{P}$ and $G$ denote the dynamic and algebraic models of *InSys*, which are readily obtained from the *InSys* physics.

Obviously, the dynamic simulation of (2) requires assembling the *InSys* physics model and the *ExSys* neural model, which is referred to as the *closed-loop simulation* of *InSys* and *ExSys* in the following discussion.

## III. Physics-Aware NeuDyE

### *A. Data-Driven Training for NeuDyE and Deficiency Analysis*

Ideally, (1) can be independently trained as follows [5]:

$$\min_\theta \sum_i \|x_{ex,i} - \hat{x}_{ex,i}\|_2, \text{ s.t. } \dot{x}_{ex} = \mathcal{N}_\theta(x_{ex}, \hat{s}_{in}) \tag{3}$$

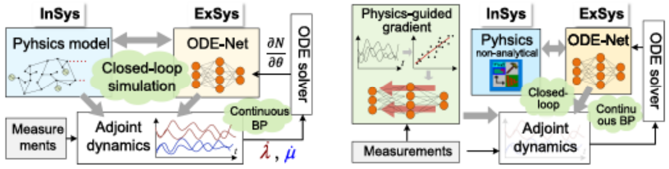
Manuscript received 25 January 2023; revised 9 May 2023 and 20 August 2023; accepted 3 October 2023. Date of publication 27 October 2023; date of current version 26 December 2023. This work was supported in part by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number 38456 and in part by the National Science Foundation under Grant OIA-2134840. Paper no. PESL-00033-2023. *(Corresponding author: Yifan Zhou.)*

Qing Shen, Yifan Zhou, and Peng Zhang are with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: qing.shen@stonybrook.edu; yifan.zhou.1@stonybrook.edu; p.zhang@stonybrook.edu).

Qiang Zhang, Slava Maslennikov, and Xiaochuan Luo are with the ISO New England Inc., Holyoke, MA 01040 USA (e-mail: qzhang@iso-ne.com; smaslennikov@iso-ne.com; xluo@iso-ne.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2023.3328162.

Digital Object Identifier 10.1109/TPWRS.2023.3328162



(a) Physics-informed (PI) NeuDyE (b) Physics-guided (PG) NeuDyE

Fig. 1. Schematic diagram of physics-aware NeuDyE.

where $i$ denotes time point $t_i$; ^ denotes the measurements.

Equation (3) minimizes the error between the measured *ExSys* states and the solution of ODE-Net (i.e., (2b)). An obvious issue is that it only considers the impact from *InSys* on *ExSys* (i.e., (2b)) while the impact from *ExSys* on *InSys* (i.e., (2a)) is ignored, leading to an *open-loop training* without involving the *closed-loop simulation*. Such an open-loop manner fails to explicitly monitor and control the closed-loop accuracy of (2) in its training process. Actually, even a tiny numerical error in the *ExSys* model can perturb the *InSys* dynamics (see (2a)). When such errors accumulate during the interactions between *ExSys* and *InSys*, the dynamic equivalence may potentially lose efficacy.

### B. Physics-Informed Learning for NeuDyE

To bridge the gap, we devise a physics-informed NeuDyE (PI-NeuDyE), which actively controls the closed-loop accuracy of NeuDyE through a physics-aware training process.

The principal idea is to leverage the well-established physics laws of *InSys* to assist the training of NeuDyE for *ExSys*. Accordingly, the following training model is developed:

$$\min_{\theta} \sum_{i=1}^{n} L_i = \sum_{i=1}^{n} \|x_{ex,i} - \hat{x}_{ex,i}\|_2 + \|x_{in,i} - \hat{x}_{in,i}\|_2 \tag{4a}$$

$$\text{s.t. } \dot{x}_{in} = \mathcal{P}(x_{in}, y_{in}, y_b)\,, \;\; \dot{x}_{ex} = \mathcal{N}_\theta(x_{ex}, s_{in}) \tag{4b}$$

As illustrated in Fig. 1(a), different from (3), the model in (4) explicitly embeds the accuracy of both *ExSys* and *InSys* states in a closed-loop manner, which ensures NeuDyE generates dependable dynamic responses in conformance with the system's real dynamics once the training converges.

Meanwhile, the training model in (4) differentiates from conventional discrete-time learning because it directly minimizes the difference between real and trained dynamic states without any discretization. This continuous-time learning manner is theoretically more invulnerable to residue training errors and non-ideal measurements, which will be further illustrated in Section IV-D1.

Training the model in (4) is not trivial. The major difficulty lies in the fact that ODE-Net's output is the derivative of $x_{ex}$ (see (4b)), but the objective targets minimizing the error between $x_{ex}$ and $\hat{x}_{ex}$ (see (4a)). Therefore, the conventional gradient descent can not be directly applied. To address the challenge, a physics-informed continuous backpropagation (BP) technique is developed to optimize (4). We leverage the adjoint method [4] to handle the differential constraints in (4b):

$$\mathcal{L} = \sum_{i=1}^{n} L_i - \int_{t_0}^{t_n} \left[\lambda^T (\dot{x}_{ex} - \mathcal{N}_\theta) + \mu^T (x_{in} - \tilde{\mathcal{P}})\right] \mathrm{d}t \tag{5}$$

where $\lambda$ and $\mu$ respectively denote the adjoint states for *ExSys* and *InSys*; $\tilde{\mathcal{P}}$ is equivalently reformulated from $\mathcal{P}$ by incorporating (2c), leading to a function of $x_{ex}$ and $x_{in}$.

Accordingly, the gradient of $\mathcal{L}$ w.r.t $\theta$ is calculate as:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta} &= \sum_{i=1}^{n} \left( \frac{\partial L_i}{\partial x_{ex,i}} \frac{\partial x_{ex,i}}{\partial \theta} + \frac{\partial L_i}{\partial x_{in,i}} \frac{\partial x_{in,i}}{\partial \theta} \right) \\ &- \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} \lambda^T \left( \frac{\partial \dot{x}_{ex}}{\partial \theta} - \frac{\partial \mathcal{N}}{\partial x_{ex}} \frac{\partial x_{ex}}{\partial \theta} - \frac{\partial \mathcal{N}}{\partial x_{in}} \frac{\partial x_{in}}{\partial \theta} - \frac{\partial \mathcal{N}}{\partial \theta} \right) \mathrm{d}t \\ &- \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} \mu^T \left( \frac{\partial \dot{x}_{in}}{\partial \theta} - \frac{\partial \tilde{\mathcal{P}}}{\partial x_{in}} \frac{\partial x_{in}}{\partial \theta} - \frac{\partial \tilde{\mathcal{P}}}{\partial x_{ex}} \frac{\partial x_{ex}}{\partial \theta} \right) \mathrm{d}t \end{aligned} \tag{6}$$

With proper adjoint boundary conditions [5], the physics-informed gradient can be yielded from (6), which includes the "adjoint dynamics" of $\lambda$ and $\mu$ (see (7a) and (7b)) and the "gradient dynamic" for $\partial \mathcal{L}/\partial \theta$ (see (7c)):

$$\frac{\mathrm{d}\lambda^T}{\mathrm{d}t} = -\lambda^T \frac{\partial \mathcal{N}}{\partial x_{ex}} - \mu^T \frac{\partial \tilde{\mathcal{P}}}{\partial x_{ex}} \tag{7a}$$

$$\frac{\mathrm{d}\mu^T}{\mathrm{d}t} = -\lambda^T \frac{\partial \mathcal{N}}{\partial x_{in}} - \mu^T \frac{\partial \tilde{\mathcal{P}}}{\partial x_{in}} \tag{7b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \left( \frac{\partial \mathcal{L}}{\partial \theta} \right) = \lambda^T \frac{\partial \mathcal{N}}{\partial \theta} \tag{7c}$$

Finally, the gradient descent for $\mathcal{N}_\theta$ can be performed using $\partial \mathcal{L}/\partial \theta|_{t=0}$ integrated from (7) by arbitrary ODE solvers.

Salient features of PI-NeuDyE over previous purely data-driven learning [4], [5] are twofold: I) Via the physics-informed model in (4), PI-NeuDyE theoretically ensures the *continuous-time* dynamic behaviors of both *InSys* and *ExSys* align with discrete-time measurements. II) Via the physics-informed gradient descent in (7), PI-NeuDyE explicitly controls the closed-loop accuracy of the dynamic equivalence.

In this letter, $\mathcal{N}_\theta$ is constructed by fully-connected neural networks. However, NeuDyE can flexibly incorporate arbitrary more advanced neural network structures [6], [7].

### C. Physics-Guided NeuDyE

One complication of PI-NeuDyE is that it requires the gradient of *InSys*'s physics models in the training process, i.e., $\frac{\partial \tilde{\mathcal{P}}}{\partial x_{ex}}$ and $\frac{\partial \tilde{\mathcal{P}}}{\partial x_{in}}$ in (7). In some applications, the analytical expression of those gradients may not be accessible, for example, when *InSys* is modeled in commercial software such as TSAT or PSS/E. Therefore, we further develop a physics-guided NeuDyE (PG-NeuDyE). As illustrated in Fig. 1(a), it leverages available *InSys* measurements and grid sparsity to estimate the gradient in the absence of analytical *InSys* models.

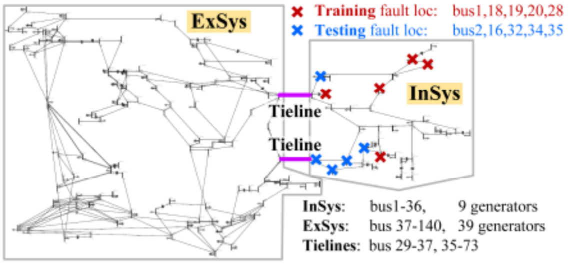

ExSys
InSys
Tieline
Tieline
Training fault loc: bus1,18,19,20,28
Testing fault loc: bus2,16,32,34,35
InSys: bus1-36, 9 generators
ExSys: bus 37-140, 39 generators
Tielines: bus 29-37, 35-73

Fig. 2. Topology of the NPCC system and fault scenario settings.

According to the Taylor expansion, the dynamic equations of *InSys* (i.e., (2a)) can be reformulated as:

$$\tilde{\mathcal{P}}_i^{(m)} = \tilde{\mathcal{P}}^{(0)} + \left[\frac{\partial \tilde{\mathcal{P}}}{\partial x_{ex}}, \frac{\partial \tilde{\mathcal{P}}}{\partial x_{in}}\right] \begin{bmatrix} \hat{x}_{ex,i}^{(m)} - \hat{x}_{ex}^{(0)} \\ \hat{x}_{in,i}^{(m)} - \hat{x}_{in}^{(0)} \end{bmatrix} + O((\Delta x)^2) \tag{8}$$

where $\hat{x}^{(m)} = [\hat{x}_{ex}^{(m)}; \hat{x}_{in}^{(m)}]$ denotes the $m^{th}$ set of time-series measurements of *ExSys* and *InSys* at time point $t_i$; $\hat{x}^{(0)} = [\hat{x}_{ex}^{(0)}; \hat{x}_{in}^{(0)}]$ denotes a typical operating point (e.g., an equilibrium point); $\tilde{\mathcal{P}}_i^{(m)} = \tilde{\mathcal{P}}(\hat{x}_{ex,i}^{(m)}, \hat{x}_{in,i}^{(m)})$; $\tilde{\mathcal{P}}^{(0)} = \tilde{\mathcal{P}}(\hat{x}_{ex}^{(0)}, \hat{x}_{in}^{(0)})$.

Denote $A$ as an estimation of $[\frac{\partial \tilde{\mathcal{P}}}{\partial x_{ex}}, \frac{\partial \tilde{\mathcal{P}}}{\partial x_{in}}]$. While the exact elements of $A$ are unknown, the sparsity structure of $A$ is usually accessible due to the connection information in power grids. Hence, the $k$-th row of $A$, i.e., $A_k$, can be estimated by the least square regression using available data samples:

$$\mathcal{S}(A_k^T) \approx \left(\mathcal{S}(X_k)\mathcal{S}(X_k)^T\right)^{-1} \mathcal{S}(X_k) p_k \tag{9}$$

Here, $\mathcal{S}(\cdot)$ denotes a sparsity transformation function, where $\mathcal{S}(A_k^T)$ extracts all non-zero elements of $A_k^T$ and $\mathcal{S}(X_k)$ extracts the corresponding elements of $X_k$. $X_k$ and $p_k$ are derived from the trapezoidal rule of (8), which respectively gather the $k$-th row of $\hat{x}_i^{(m)} + \hat{x}_{i-1}^{(m)} - 2\hat{x}^{(0)}$ and of the remained terms of (8) for arbitrary $m$ and $i$.

Accordingly, an estimation of $A$ can be recovered from $\mathcal{S}(A_k^T)$ $(\forall k)$, and, in this way, supports the implementation of the physics-informed gradient calculation in (7) if the analytical gradients are unavailable.

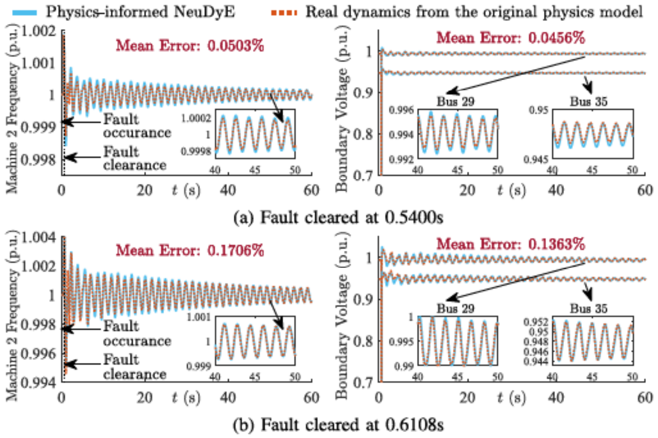
## IV. Case Study

This section performs case studies on the Northeast Power Coordinating Council (NPCC) system (see Fig. 2). The New England system (i.e., buses 1–36) is considered as the *InSys*, and the rest is the *ExSys* to be learned by NeuDyE. All codes are developed and implemented in Matlab R2022b.

### A. Experiment Settings

As shown in Fig. 2, 20 training scenarios are generated by short-circuits occurring at bus 18, 19, 20, 21, or 28 with different fault clearing times. 108 testing scenarios are generated with new fault locations and random fault clearing times at bus 2, 5, 9, 16, 25, 28, 32, 34 and 35.

The electromechanical simulations of the NPCC system are performed via the Power System Toolbox (PST). 27 generators are formulated by the electromechanical model and 21 generators are formulated by the voltage-behind-transient-reactance model, which align with the Transient Security Assessment Tool (TSAT) model provided by ISO NE. The trapezoidal rule is adopted for the numerical integration of NeuDyE.



(a) Fault cleared at 0.5400s

(b) Fault cleared at 0.6108s

Fig. 3. Accuracy of PI-NeuDyE under a trained fault location (i.e., bus 21).

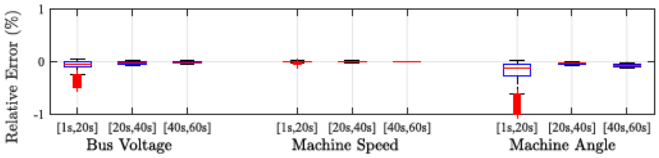


Fig. 4. Accuracy of PI-NeuDyE under new fault locations.

To learn the NeuDyE model of *ExSys* shown in (1), this letter selects the states of generators, exciters, governors, and line currents of *InSys* as $s_{in}$ (i.e., features of *InSys*), and the tieline currents as $x_{ex}$ (i.e., the states of *ExSys*). However, such training features can be flexibly adjusted according to available measurements.

### B. Efficacy of Physics-Informed NeuDyE

*1) Accuracy:* We first validate the accuracy of PI-NeuDyE. Fig. 3 presents the performance of PI-NeuDyE under short-circuit faults at bus 21 (a trained fault location) but with random fault clearing times. Trajectories of boundary bus voltages (i.e., bus 29 and bus 35) and machine frequencies of the New England grid show a perfect match between NeuDyE results (see blue lines) and real NPCC dynamics (see red lines), which illustrates the accuracy of the developed method.

Another observation from Fig. 3 is the low-damping oscillation, which is induced by the inter-area modes of the NPCC system. The efficacy of NeuDyE in capturing both the fast oscillations and the slow damping tendency demonstrates its applicability in stiff systems with multi-time-scale dynamics.

*2) Generalization Capability:* Fig. 4 quantitatively studies the generalization capability of PI-NeuDyE with new fault locations and fault clearing times. The time-series relative error of typical system states shows that even under unforeseen faults occurred at new locations, PI-NeuDyE maintains reasonable error rates along the time horizon, which indicates the efficacy of PI-NeuDyE to preserve the dynamic behaviors of *ExSys*

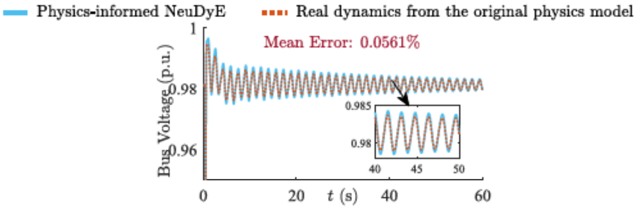
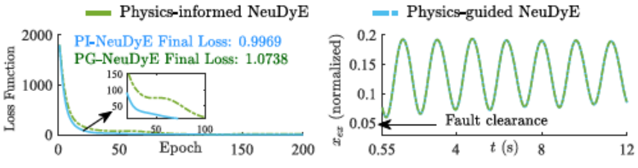
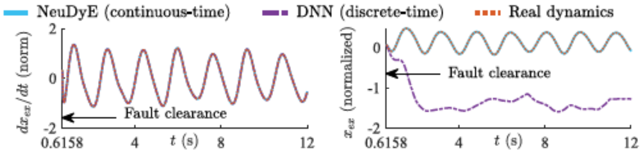


Fig. 5. Efficacy of PI-NeuDyE in parametric systems.



(a) Loss function evolution comparison (b) Dynamic trajectory comparison

Fig. 6. Performance of PG-NeuDyE and its comparison with PI-NeuDyE.



(a) Open-loop training performance (b) Closed-loop testing performance

Fig. 7. Comparison of NeuDyE with conventional discrete-time DNN.

after contingencies and its satisfactory generalization capability beyond the training set.

*3) Performance in Parametric Cases:* We further demonstrates the efficacy of PI-NeuDyE under system parameter changes. Without loss of generality, load change is selected as an impact factor to retrain the ODE-Net model, where the load level of *InSys* randomly changes from 70% to 130% of the original load level. Fig. 5 presents the test results of a specific case where the power load increases by 28% in fault conditions. The relatively low mean error illustrates the accuracy of the method in parametric dynamic systems.

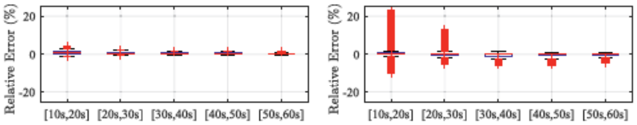
### *C. Validity of Physics-Guided NeuDyE*

Fig. 6 presents the performance of PG-NeuDyE. As shown in Fig. 6(a), although PG-NeuDyE converges slightly slower than PI-NeuDyE in the training process (because it does not use any information from the analytical physics models of *InSys*), it finally reaches a comparable loss level compared with PI-NeuDyE. Meanwhile, Fig. 6(b) shows that the dynamic equivalence obtained from PG-NeuDyE produces nearly identical dynamic behaviors with that from PI-NeuDyE, which indicates the validity of NeuDyE.

### *D. Comparison With Existing Methods*

Finally, this subsection compares the devised method with existing methods to reveal its necessity and superiority.

*1) Necessity of the Continuous-Time Consideration:* Fig. 7 compares NeuDyE with conventional deep neural network (DNN) methods to demonstrate the necessity of developing continuous-time learning-based NeuDyE. Fig. 7(b) shows that DNN fails to provide qualified dynamic responses in the closed-loop simulation despite a perfect training accuracy in Fig. 7(a).



(a) Error of PI-NeuDyE (b) Error of purely data-driven method

Fig. 8. PI-NeuDyE's error of tie-line currents under 10 fault cases.

The fundamental reason is that conventional DNN methods can not directly handle the continuous-time differential equations in (4). Therefore, they usually rely on discrete-time learning to obtain a *discretized* dynamic equivalence, whose performance is very sensitive to the training errors.

*2) Necessity of the Physics-Aware Consideration:* Fig. 8 compares our physics-aware NeuDyE with purely data-driven methods to demonstrate the necessity of incorporating physics knowledge into NeuDyE. The relatively large error in Fig. 8(b) indicates the purely data-driven dynamic equivalence does not guarantee its closed-loop accuracy even though the model is well trained. This observation aligns with the discussion in Subsection III-A because purely data-driven methods can not explicitly control the closed-loop accuracy in their training. In contrast, physics-aware NeuDyE actively incorporates the bi-directional interactions between *InSys* and *ExSys*, and hence supports controllable closed-loop accuracy for NeuDyE.

## V. Conclusion

This letter devises physics-aware Neural Dynamic Equivalence (NeuDyE), a novel technique to discover a *continuous-time* dynamic equivalence of external systems leveraging *physics knowledge* of internal systems. The most salient feature is its capability of retaining the continuous-time dynamic natures of power grids and its active control of the closed-loop accuracy during training. Case studies in the NPCC system show the efficacy of NeuDyE under various fault locations, fault clearing times, parameter change, and its superiority over existing discrete-time learning or purely data-driven methods.

## References

[1] M. L. Ourari, L.-A. Dessaint, and V.-Q. Do, "Dynamic equivalent modeling of large power systems using structure preservation technique," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1284–1295, Aug. 2006.

[2] Y. G. I. Acle, F. D. Freitas, N. Martins, and J. Rommes, "Parameter preserving model order reduction of large sparse small-signal electromechanical stability power system models," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 2814–2824, Jul. 2019.

[3] I. Tyuryukanov, M. Popov, M. A. M. M. van der Meijden, and V. Terzija, "Slow coherency identification and power system dynamic model reduction by using orthogonal structure of electromechanical eigenvectors," *IEEE Trans. Power Syst.*, vol. 36, no. 2, pp. 1482–1492, Mar. 2021.

[4] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[5] Y. Zhou and P. Zhang, "Neuro-reachability of networked microgrids," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 142–152, Jan. 2022.

[6] T. Qin, Z. Chen, J. D. Jakeman, and D. Xiu, "Data-driven learning of nonautonomous systems," *SIAM J. Sci. Comput.*, vol. 43, pp. A1607–A1624, 2021.

[7] G. Lin, C. Moya, and Z. Zhang, "On learning the dynamical response of nonlinear control systems with deep operator networks," 2022, *arXiv:2206.06536*.