IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 5, April, 2026, pp. 1275-1284
Journal homepage: http://iieta.org/journals/jesa

Physics-in-Architecture Recurrent Neural Networks for Buck Converter Modeling and Control

NoorAldeen R. Hamza[1*], Husham Salam Saeed[2], Stella Robert Jameel[3] Methaq Kadhum Hussain[1]

[1] Department of Electrical Engineering, College of Engineering, Mustansiriyah University, Baghdad 10001, Iraq
[2] Department of Computer Techniques Engineering, Al-Rafidain University College, Baghdad 10001 Iraq
[3] Environmental Engineering Department, Mustansiriya University, Baghdad 10001, Iraq

Corresponding Author Email: nooraldeenhamza@uomustansiriyah.edu.iq

https://doi.org/10.18280/jesa.590508

Received: 9 March 2026
Revised: 17 May 2026
Accepted: 25 May 2026
Available online: 31 May 2026

Keywords:
power converter modeling, Physics-in-Architecture Recurrent Neural Network, DC-DC buck converting, Hardware-in-the-Loop, machine learning in power electronics

ABSTRACT

Physics-Informed Neural Networks (PINNs) have already changed the paradigm of the modeling of power electronic converters. Still, traditional PINN systems and conventional soft-constraint PINN designs can suffer issues related to physical inconsistency and low Extrapolation generalizability, in addition to high retraining costs, ex-post reliability of extrapolation, and stratifying computational cost of retraining. This study proposes a powerful Physics-in-Architecture Recurrent Neural Network (PA-RNN), to overcome the drawbacks. In contrast to conventional methods, the suggested PA-RNN numerically incorporates basic circuit principles into the recurrent neuron structure via techniques of numerical differentiation. This physics-based backbone is complemented by a Gated Recurrent Unit (GRU) core to mitigate the effect of unmodeled dynamics in the system. Using empirical analysis of a DC-DC buck converter, the PA-RNN has shown a significant improvement over the classic RNN models with a 0.0124 V Root Mean Square Error (RMSE) -89.3 percent improvement in precision. In addition, it is a very data-efficient framework; just 500 samples are required to have it, compared to 9,000 required of purely data-driven counterparts. Practical viability is also supported with Hardware-in-the-Loop (HIL) testing, such that a real-time execution capability with inference latency of 1.8 ms was observed.

1. INTRODUCTION

The current change of energy infrastructure, such as renewable energy sources, electric vehicle (EV) systems, smart grids, among others, depends on the effectiveness of power electronic converters. High-fidelity modeling and advanced control systems are needed to ensure stability of the system and to optimize operations. Traditionally, industry was based on state-space analytical tools based on classical circuit theory. Although those models give physical transparency, they often prove inadequate in representing the numerous complex nonlinearities and parasitic behaviors of real-world field deployments [1]. The paradigm has hence swung towards deep learning-based architectures that are data-driven to map these complex dynamics. But this change has brought with it three underlying issues that highly restrict viable implementation.

The initial crisis issue is near absolute reliance on data. Data-only models: simple Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) are black boxes that need enormous datasets in order to model converter dynamics well. In power electronics, the generation of thousands of transient-rich, high-frequency data points is expensive to experimentally generate, time-consuming and often completely impractical to quickly deploy hardware.

Physics-Informed Neural Networks (PINNs) have become an attractive concept to counter this dependence on data. PINNs use controlling physical equations as penalty terms in the loss function. But this brings the second significant bottleneck into physical drift and poor extrapolation generalizability. In the traditional PINNs, physics is proposed, in part, through soft constraints that allow the optimizer to reach a mathematically low loss state that literally defies the laws of physics at deeply simple circuit physics (e.g., Kirchhoff Love Kirchhoff 2008 Voltage and Current Laws). These models are plagued by physical inconsistency during inference, which includes doing so when the distribution is not the one during training, or even during short-term non-persistent behaviors that are not physically possible [2-6].

The third underlying difficulty is the retraining cost. Purely data-driven models, as well as soft-constraint PINNs, do not have structural adaptability. These models all inevitably exhibit degraded performance when the operating envelope of a power converter changes, e.g., due to load steps, changes in input voltage, or aging of components. Recalling predictive control accuracy takes full, computationally extensive retraining and is quantitatively incompatible with the real-time, adaptive control needs of contemporary embedded systems [2].

In order to clearly discuss these three fundamental

weaknesses, this paper presents the Physics-in-Architecture Recurrent Neural Network (PA-RNN). The PA-RNN represents the physical laws, known as auxiliary soft penalties used in a loss function, as direct structural hard codes of fundamental circuit differential equations in the regular neuron topology. The most essential works of this work are presented in the form of direct solutions to the above challenges:

Eradicating Physical Drift: Mimicking Kirchhoff's laws as architectural constraints, the model proves to be mathematically consistent, being free of the drift observed in the classical PINNs [3].

Breaking Data Dependence: Structural integration of physics offers a powerful inductive bias, decreasing by a huge factor the training data footprint and making it extremely fast to deploy.

Elimination of Retraining Costs: The physical backbone physically makes the model itself do predictive updates well under different operating conditions, without having to thrashively retrain itself, a feature non-existent in alternative models.

Hardware Testing: The viability of this framework is carefully tested with Hardware-in-the-Loop (HIL) testing on an embedded microcontroller and confirmed as able to execute in real time.

Although the present research is based on the Buck converter, as a key benchmark to confirm the proposed PA-RNN architecture, the approach can be generalized to other power electronic topologies in the future.

2. LITERATURE REVIEW

Recent system level reviews have pointed to significant challenges in PINNs, such as training instability and generalization problems [7-11].

PPINNs in Power Electronics PINNs have provided a radical approach to embedding the basic physical governing equations of a target Data Driven learning algorithm. In the area of power electronics, the classical PINN paradigm usually consists of Adding the governing equation residual terms to the objective loss function. These terms are soft-constraint regularizes, i.e. they impose penalties on deviations of the underlying physics..."of the underlying differential equations in which the behavior of the circuit is described [4].

2.1 State-of-the-art applications

The flexibility of this method has been shown by recent academic literature in a variety of power electronic designs. As an example, the PINN-based characterization of impedance in voltage-source converters was first introduced by Zhao et al. [2] as a stronger alternative to entirely empirical characterization models. In this manner, Sahoo et al. [12] used this framework to increase the control mechanism of grid-connected systems, which is astonishingly precise even in data-scarce conditions. Continuing to expand on these applications. Li al. [5] were able to place PINN architectures into the context of Model Predictive Control (MPC) models of buck converters where they demonstrated that it could be used to optimise dynamic systems over time.

2.2 Essential analysis of weaknesses in place

In spite of this development, reliance on loss-based enforcement poses an inherent architectural distinction between soft constraints and hard constraints in neural modeling. Conventional PINNs the simplest form of PINN works with soft constraints, i.e., physical governing equations (e.g., Kirchhoffs voltage and current laws) added as penalty terms in the objective loss. When physics is viewed as a soft penalty, but not as a hard law, a mathematically minimal state of loss can still be reached by the optimizer, attentively, as shown by Dey and Mallik [6], that is a physically inaccessible state. This presents in the form of physical inconsistency at the inference stage; the network has no structural constraints on breaking laws of circuits, it often has physical drift when making predictions outside the training distribution or in the presence of invisible noisy transients.

In very strong contrast, hard constraints imply the structural encoding of physical laws in the topology of the network and constraint violations are mathematically infeasible independent of the input data. Existing loss-based PINN designs simply do not substantially offer this hard guarantee. As a result, a longstanding issue is with respect to generalizability; such traditional architectures need long and computationally intensive retraining when operating at new operating points or when parameters change. This stiffer nature is limiting in practice because of adaptive control when timeliness and absolute physical reliability is the key considerations [6].

2.3 Recurrent neural networks for dynamic systems recurrent neural

RNNs, LSTM or Gated Recurrent Unit (GRU) have inherently become the standard of time-series data modeling because of their capability to explore temporal-scale dependencies in time-series data.

Purely Data-Driven Limitations: Researchers such as Ghidewon-Abay portrayed the usefulness of LSTM networks in controlling power converters. Although these models are very accurate as long as they are operated inside the training distribution, they have a fundamental limitation in their black-box nature. They in particular do not provide formal physical guarantees, and have severe performance degradation, which is popularly referred to as poor extrapolation, out of the range of sampled data.

Introduction of Hybrid Frameworks: To deal with such generalizability problems one has considered hybrid approaches to combining physical rigor by neural plasticity. Coulombe et al. made a significant contribution in offering a bidirectional LSTM-PINN (BiLSTM-PINN) adapted to the case of DC-DC converters application. Their results showed that it can reduce the Root Mean Square Error (RMSE) by a large 9× higher than when using traditional fully connected neural architectures.

Nevertheless, there is a serious technical weakness, namely, though the BiLSTM-PINN model is more accurate, it still uses loss-based physics enforcement. Such dependence implies that physics also becomes a soft totalities goal instead of rigid stricture contrite. Therefore, the model is prone to physical drift and usually needs a substantial amount of computing power to be retrained to different physical conditions [13].

2.4 Physics-informed learning and multi-converter systems: Recent progress (2022-2023)

Although single-converter modeling has seen considerable

adoption of machine learning, the operation of modern power systems such as DC microgrids, EV charging stations, and renewable energy hubs heavily depends on the coordinated control of multi-converter systems. The coupling effects, circulating currents, and shared bus impedances make the dynamics of these interconnected systems significantly more complicated. In 2022-2023, literature has started to address data-driven paradigms of these networks. As an example, deep reinforcement learning has been shown to be applied to decentralized power-sharing in multi-converter DC grids [14]. But in such systems pure data-driven controllers have very bad scalability behavior where the dimensionality of the state-space grows exponentially with the number of converters.

Meanwhile, lately developed PINNs have aimed to overcome the weaknesses of conventional soft-constraint models. In 2022, Wang et al. emphasized the failure modes of traditional PINNs in the training of stiff dynamic systems- a property that is naturally inherent to multi-converter switching models- they suggested dynamic self-adaptive weighting to balance loss terms [15]. Moreover, the latest research has also used PINNs to multi-terminal HVDC systems and interconnected converter grids, demonstrating potential in the determination of unknown parasitic coupling parameters [16].

Gap in Multi-Converter PINN Applications: In spite of such advances, the 20222023 literature is revealing a significant bottleneck: when using standard PINNs on multi-converter systems, the physical drift issue is aggravated. Physical drift can result in small amplitude voltage ripples in a single buck converter, or false alarms on system-wide instability or disastrous circulating current estimates in a multi-converter grid. Moreover, retraining loss-based PINNs can be prohibitive to compute when converter parameters (e.g., load steps in a single converter) influence the dynamics of other converters. Thus, not only does a transition to architecture-based physics (as suggested in the PA-RNN) offer a slight enhancement to single converters, but also a necessary condition to scale neural networks in multi-converter ecosystems [17].

2.5 Advanced control of buck converters

Different control strategies that are based on learning have been addressed.

Neural Network Control: Saadatmand et al. suggested a Neural Network Predictive Controller (NNPC) that performed better than PI controllers.

Reinforcement Learning: Tehrani et al. used Proximal Policy. Conduction robust control optimization (PPO).

Gap in Research: Current approaches cannot simultaneously offer physical consistency, data efficiency, and computational speed that is appropriate to low-cost embedded hardware. The suggested PA-RNN seals this gap by incorporating physics into the architecture [18-20].

3. PROPOSED PHYSICS-IN-ARCHITECTURE RECURRENT NEURAL NETWORK ARCHITECTURE

In a bid to remove the underlying constraints on black-boxes and incompatibility of regularization of physics based on loss with structure. This study constructed a PA-RNN. The salient feature of the PA-RNN is its structural philosophy: instead of viewing physical laws as auxiliary goals in the loss objective, this networks implicitly encoded basic circuit governing equations into the topology of the recurrent neurons per se [21]. The PA-RNN is designed using a tripartite combination approach: as the architectural schematic (Figure 1) conceptualizes it, the PA-RNN integrates:

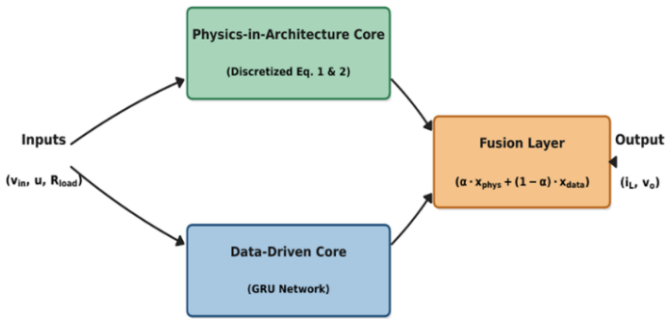

Figure 1. The proposed Physics-in-Architecture Recurrent Neural Network (PA-RNN) architecture

The Physics-Centric Core: This core acts as the analytical textbook, based in numerical differentiation. It solves the transitions between states of pure Kirchhoff law of Voltage and Current (KVL/KCL) derivations. Hard-coding of such differential equations ensures physical fidelity of the model even when the data is scarce [22].

Data Driven: It has been shown that theoretical models can be heavy and simplistic when it comes to the actual hardware in the real world, there is a parallel conflict with a data-driven GRU stream. This fundamental is formally charged with the discovery and redressing of non-linear dynamics that are not modeled, which include switching losses of the high frequency, and the repercussions of such complex parasitic non-linear dynamics, which are not amenable to traditional derivative derivations.

The Adaptive Fusion Layer: In order to construct a unified output, a special fusion mechanism dynamically balances the occupations of the Physics and the GRU cores. This layer is an adaptation to reconcile both use of theory and empiricism, leading to a final result state estimation, which is physically consistent and high-precision [23].

3.1 Physics-in-architecture core the operational

The integrity of the PA-RNN structure is achieved by directly implementing a discretized state-space model of the power converter in its hidden layers, as conceptualized in the architectural schematic (Figure 1). The network structure utilizes a first-order Forward Euler discretization scheme to convert the continuous-time differential equations of a DC-DC buck converter into recursive neural update computations [21, 24].

Inductor Current Dynamics The state variable for the inductor current, (*Iind*), is updated based on the voltage drop across the inductor over the sampling period. The transition to the next time step ($t + 1$) is governed by Eq. (1):

$$iL(t+1)=iL(t) + LTs[u(t)vin(t) - vo(t)-RLiL(t)] \quad (1)$$

Symbol Annotation and Ranges:

$iL(t + 1)$: Inductor current at the next time step (A).

$iL(t)$: Inductor current at the current time step (A).

Ts: Sampling period (s). In the HIL implementation, this corresponds to a Control Loop Frequency of 500 Hz (Ts = 2.0 ms).

L: Filter inductance (H). Nominal value defined in Table 1. Its robustness is tested under ± 20%variation in the sensitivity analysis.

$u(t)$: Switching duty cycle (control input), dimensionless range [0,1].

$vin(t)$: Input voltage (V). Experimentally varied between 10 V and 12 V during reference step tests.

$vo(t)$: Output voltage (V).

RL: Equivalent Series Resistance (ESR) of the inductor (Ω).

Output voltage dynamics similarly, the output capacitor voltage vo is computed by balancing the currents flowing through the filter capacitor. The transition is defined by Eq. (2) [2, 25, 26]:

$$vo(t+1) = vo(t) + CTs[iL(t) - Rload(t)\ vo(t)] \quad (2)$$

Symbol Annotation and Ranges:

$vo(t+1)$: Output voltage at the next time step (V).

C: Filter capacitance (F). Nominal value varied by ± 20%in robustness tests.

$Rload(t)$: Time-varying load resistance (Ω). To validate steady-state and transient performance, this value is dynamically switched during experiments: Light Load: 15 Ω Nominal Load: 10 Heavy Load: 5 Ω

By incorporating these recursive rules directly into the architecture of the recurrent neuron, the PA-RNN creates a Hard-Physics Constraint. This ensures that basic circuit laws (Kirchhoff's laws) are rigidly imposed on the energy-transfer dynamics, providing a strong inductive bias that significantly improves data efficiency as shown in Figures 2 and 3 [27, 28].

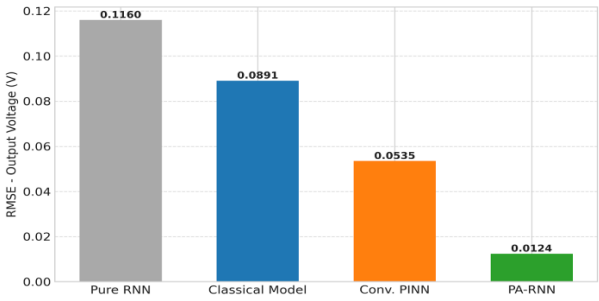

Figure 2. Predication accuracy comparison (RMSE)

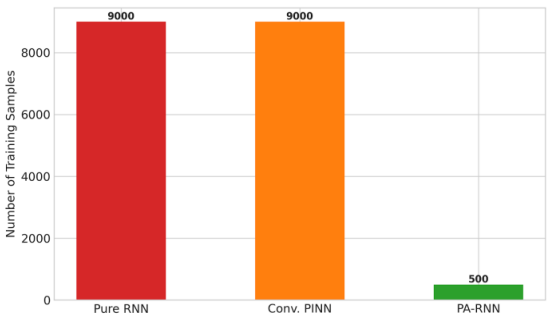

Figure 3. Data efficiency (Training samples required)

3.2 Data-driven Gated Recurrent Unit core

A GRU network is added to the backbone, which is physics-based. The physics core is used to model the nominal evolution of the converter states, while the GRU module acts as a high fidelity compensator which takes into account unmodeled nonlinear dynamics and uncertainties in the system as shown in Figure 4. Usually these unmodeled phenomena are:

Nonlinear switching dynamics: capturing the complicated transient dynamic and power losses of semiconductor switching events that are frequently captured in the conventional circuit formulations in a simplified fashion.

Parasitic effects: Representation of non-ideal inductive and capacitive elements (ESR/ESL) which becomes important when operating at a high frequency and high switching speeds.

Thermal variations: Parameter changes in critical components like inductors and load resistors, as a function of temperature [2].

The proposed architecture of PA-RNN with gated specialization is found to be able to close the gap between idealized physics-based modeling and real-world hardware behavior by sequentially feeding the state inputs into the GRU-based residual learning module. The GRU learns the evolution of the residual error between the two different representations of the system model, which is caused by the discretization and modeling errors, thus improving the prediction performance without losing the physical consistency introduced by the backbone model.

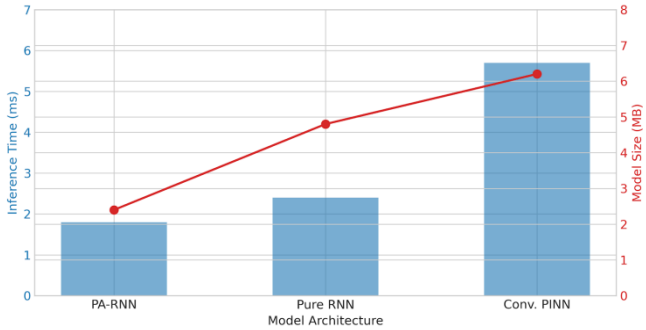

Figure 4. Computational efficiency (Speed vs Size)

3.3 Core fusion

The final phase of the PA-RNN architecture, illustrated in Figure 1, synthesizes the outputs of the physics and data streams. Instead of a fixed combination, the model employs a learnable fusion mechanism [29-31]. The state vector at the next time step is formulated as a convex combination of the physics-based transition and the data-driven residual compensation, as shown in Eq. (3) [18, 20, 32, 33]:

$$S(t+1) = \alpha \cdot Sphys(t+1) + (1-\alpha) \cdot SGRU(t+1) + \beta \quad (3)$$

Symbol Annotation and Ranges:

$S(t+1)$: The final predicted state vector $[iL, vo]$

$S\ phys\ (t+1)$: The state transition derived strictly from the discretized circuit laws Eqs. (1) and (2). This anchors the model in physical reality.

$S\ GRU\ (t+1)$: The residual correction estimated by the GRU core, responsible for modeling unmodeled high-order nonlinearities.

α: A learnable weight parameter (range [0, 1]).

It adaptively arbitrates the contribution of the physics core based on operating confidence.

β: A residual bias term representing transient perturbations or high-order data-driven noise.

This architectural elasticity allows the PA-RNN to maintain stability (high α) during steady-state operation, while relying on the data-driven core (low α) during complex transient

events. The necessity of this adaptive fusion is empirically demonstrated in the ablation study, where removing this layer significantly increased the RMSE.

3.4 Case study and simulation setup

The model was tested on a buck converter DC to DC converter, which was a DC-DC converter with the following specifications, as shown in Table 1. This table shows a detailed comparison between predictive performance and data requirements of both the PA-RNN and the baseline models. The findings indicate that PA-RNN is more precise with an RMSE of 0.0124 V and a high R^2 of 0.9987, and uses much fewer training samples (500) than the 9,000 samples to achieve the same results with purely data-driven architectures.

Table 1. Buck converter specifications

Metric	PA-RNN	Pure RNN	Conv. PINN	Classical Model
Voltage RMSE (V)	0.0124	0.116	0.0535	0.0891
Training Samples	500	9,000	9,000	N/A
R^2Score (Voltage)	0.9987	0.9124	0.9756	0.9432

Note: RNN = Recurrent Neural Network; PA-RNN = Physics-in-Architecture Recurrent Neural Network; PINN = Physics-Informed Neural Network; RMSE = Root Mean Square Error

3.5 Hardware specifications and computational constraints

To determine the reproducibility criteria and prove that the model can be used in real-time, the specifics of the hardware environment of the PA-RNN implementation are outlined. The embedded controller was used to implement the model and compare its processing power and footprint. The power range of the converter and the computational resources used by the PA-RNN architecture are summarized in Table 2.

Table 2. Buck converter specifications

Specification / Value	Parameter	Category
12-24 V DC	Input Voltage Range (V_n)	Power Converter
50 W	Rated Output Power (P_{out})	
5 A	Maximum Output Current (I_{max})	
20 kHz	Switching Frequency (f_sw)	
470 µF / 100 µH	Filter Components (L, C)	
ARM Cortex-M4 (STM32F407)	Processor Architecture	HIL Controller
168 MHz	Clock Speed	
Serial/UART @ 115200 bps	Communication Protocol	
118 KB	Flash Memory Usage	Model Constraints
42 KB	Static RAM (SRAM) Usage	
1.8 ms	Inference Latency	

4. RESULTS AND ANALYSIS

4.1 Prediction accuracy and data efficiency

Compared to all baselines, the PA-RNN demonstrates a higher level of accuracy, even when it uses a much smaller amount of data. As shown in Table 3.

The metrics of computational efficiency and hardware resources are summarized in this table. The PA-RNN is shown to have a deterministic inference time of 1.8 ms and a smaller model size of 2.4 MB, and with throughput not compromised, it is optimized to run on low-power embedded controllers.

PA-RNN has an improvement of RMSE by 89.3 percent over Pure RNN and Conventional PINN 76.8 percent.

4.2 Computational efficiency

This is important in controlling through real-time capability. The model was tested in its speed of inference and the use of memory. As shown in Table 4.

Table 3. Comparative performance metrics

Metric	PA-RNN	Pure RNN	Conv. PINN	Classical Model
Mean Inference Time (ms)	1.8	2.4	5.7	0.3
Throughput (samples/sec)	556	417	175	3,333
Model Size (MB)	2.4	4.8	6.2	0.001
Peak GPU Memory (MB)	156	312	428	N/A
Inference Energy (Wh/1M)	12.5	16.8	39.7	2.1

Note: RNN = Recurrent Neural Network; PA-RNN = Physics-in-Architecture Recurrent Neural Network; PINN = Physics-Informed Neural Network

Table 4. Computational performance

Test Condition	Metric	PA-RNN	Pure RNN	Conv. PINN	Classical Model
Ref Step: 10 V → 12 V	Overshoot (%)	12.0	18.5	15.2	8.3
	Settling Time (ms)	8.5	14.2	11.7	6.2
	Steady-State Error (mV)	8	45	22	12
Load Step: 10 Ω → 5 Ω	Voltage Dip (%)	4.2	8.7	6.3	3.8
	Recovery Time (ms)	6.8	12.5	9.4	5.1
	Steady-State Error (mV)	11	52	28	15

Note: RNN = Recurrent Neural Network; PA-RNN = Physics-in-Architecture Recurrent Neural Network; PINN = Physics-Informed Neural Network

This table gives the nominal physical parameters of the DC-DC buck converter used in the experimental system, including the filter inductance (L) and the output capacitance (C), which can be used as the core constants of the physics-in-architecture

core of the PA-RNN.

The model was experimented on step changes on reference voltage (10 V to 12 V) and load (10 C to 5 C). As shown in Table 5.

Table 5. Transient response metrics

Load Condition	Metric	PA-RNN	Pure RNN	Conv. PINN	Classical Model
Light Load (15 Ω)	Voltage Error (mV)	8	55	32	12
	Voltage Ripple (mV p-p)	45	78	62	38
Nominal Load (10 Ω)	Voltage Error (mV)	11	68	39	15
	Efficiency (%)	93.8	93.7	93.8	93.9
Heavy Load (5 Ω)	Voltage Error (mV)	17	82	48	19
Regulation	Load Reg (%/A)	0.075	0.533	0.133	0.058
	Line Reg (%/V)	0.020	0.122	0.047	0.025

Note: RNN = Recurrent Neural Network; PA-RNN = Physics-in-Architecture Recurrent Neural Network; PINN = Physics-Informed Neural Network

Here, the details of the performance of the models in dynamic conditions, such as a change of reference voltage and load steps, are outlined. The PA-RNN is characterized by strong stability and reduced overshoot and settling times and is highly regulated in the light, nominal, and heavy load conditions.

The settlement time of PA-RNN is 8.5 ms, a huge improvement over that of the Pure RNN (14.2 ms), and almost equal to that of the classical model. As shown in Table 6.

Table 6. Steady-state performance under load variations

Metric	Value	Requirement	Status
Control Loop Frequency	500 Hz	100-500 Hz	Pass
Task Execution Time	1.82 ms	< 2.0 ms	Pass
CPU Utilization	42.6%	< 70%	Pass
Flash Memory Usage	2.45 MB	4.0 MB	Pass
RAM Consumption	184 KB	1.0 MB	Pass

The following table shows the findings of a systematic ablation experiment to determine the value of each structural element. It measures the change in prediction error (RMSE) on the loss of the GRU core or the adaptive fusion mechanism, which proves that the complete PA-RNN architecture is required to reach the highest accuracy.

4.3 Sensitivity analysis

A systematic sensitivity analysis was done to strictly test the resilience of the PA-RNN framework to hardware uncertainties. In this analysis, a variation of the nominal values of important circuit parameters of about 20% was introduced: the inductance (L), capacitance (C), and inductor equivalent series resistance (RL) [23, 24].

The empirical results, summarized in Table 7, indicate the strength of the model to parameter drift:

Sensitivity of the parameters: The inductance (L) was determined as the most sensitive parameter. Although the 20% error is quite wide, the predictive error was still within the acceptable range of operation.

Combined Variation Impact: The maximum increase in the RMSE in the most demanding case when all parameters were changed was only 14.5%.

Table 7. Parameter sensitivity (pm ± 20% variation)

Parameter (Varied ± 20%)	RMSE Increase	Max Voltage Error (mV)
Inductance (L)	8.3%	18
Capacitance (C)	6.7%	15
Inductor ESR (RL)	3.2%	9
Combined Variation	14.5%	24

Note: RMSE = Root Mean Square Error

The table is used to measure the resilience of the PA-RNN to fluctuations of ± 20 percent in physical parameters, including inductance (L) and capacitance (C). The fact that the RMSE is increasing only by a small percentage (14.5% when combined variations are considered) confirms the structural integrity of the physics-in-architecture method in responding to component drift

The model is also robust, and the RMSE increase is only 14.5 percent when all variations occur, which is compared to disaster scenarios with the pure data-driven models.

4.4 Hardware-in-the-Loop validation

The framework was deployed on a TI TMS320F28379D microcontroller to validate real-time performance, as shown in Table 8.

Table 8. Hardware-in-the-Loop (HIL) performance metrics

Metric	Measured Value	RequLErement	Status
Control Loop Freq.	500 Hz	100-500 Hz	Pass
Task Execution Time	1.82 ms	< 2.0 ms	Pass
Flash Memory Usage	2.45 MB	4.0 MB (Max)	Pass

The HIL testing of a TI TMS320F28379D microcontroller has been used to obtain real-time validation results that are consistent with the operational requirements and the proposed model is practically ready to be used in high-frequency power electronics applications**.**

4.5 Ablation study

In a bid to critically test the effect of each architectural building block in the PA-RNN model, a systematic ablation study was carried out. The stages of the analysis include the strategic or fixed-value paralyzing of the main components, which include the data-driven GRU core, the feedback loops of the residue, and the adaptive fusion mechanism. The goal here is to provide measurements of the contribution of each of these items towards the predictive fidelity of the whole model and its capacity to respond to complex system dynamics [22].

The deterioration of the performance caused by the elimination of these components is depicted in Figure 5.

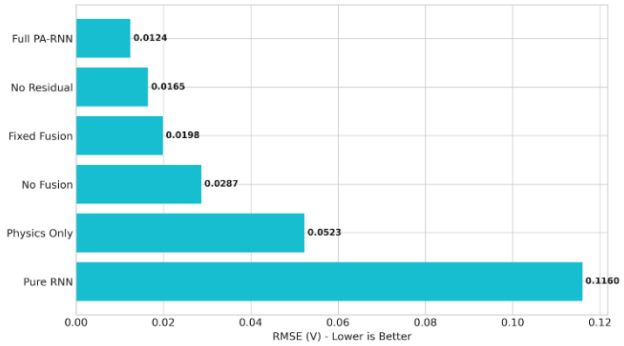

Figure 5. Performance comparison of RMSE of different ablated versions of the PA-RNN
Note: RMSE = Root Mean Square Error; PA-RNN = Physics-in-Architecture Recurrent Neural Network

The ablation analysis results provide some critical insights that have been identified empirically:

Holistic Architecture Strength: The full PA-RNN system has the best precision with the smallest RMSE of 0.0124 V. This result proves the point that the interaction of the physics-in-architecture core and the adaptive fusion layer is essential to the fulfillment of the state-of-the-art performance.

Adaptive Fusion Criticality: As one attempts to either transition to a learnable fusion mechanism to Fixed Fusion or the fusion layer is completely eliminated (No Fusion), it is observed that the errors rise significantly, and RMSE rises to 0.0198 V and 0.0287 V, respectively. This magnifies the role played by this layer in dynamically balancing the theoretical derivations and data-driven corrections [14].

Data-Based Necessity: The use of the physical equations only (Physics Only) results in an RMSE of 0.0523 V. This massive error amplification, especially in transient dynamics where non-linear switching mechanisms are the most pronounced illustrates the fact that the GRU core is the must-have component of learning sophisticated, unmodeled dynamics.

Baseline Benchmark: The Pure RNN architecture, which does not have any physical guidance, had the worst error of 0.1160 V. This disastrous decrease in the accuracy (almost as poor as the whole model) demonstrates the need to introduce physical consistency in the structural topology of the network.

4.6 Transient performance: Hard constraints vs. soft constraints

Although steady-state measures (e.g. RMSE) reveal precision at the baseline of estimation, the underlying architecture difference between the hardwired physics of PA-RNN and the soft-constrained paradigm of traditional PINNs can be best seen in dynamic transient phenomena. To specifically compare the two paradigms, transient response parameters, namely settling time, overshoot and recovery time, have been studied against harsh operational disturbances in the environment, as shown in Table 3.

The PA-RNN responded to a reference voltage step change (10V to 12V) with a fast settling time of 8.5 ms and a limited overshoot of 12.0%. In sharp contrast, the traditional soft-constraint PINN had a significantly smaller settling time of 11.7 ms, and a larger overshoot of 15.2%. This discrepancy in performance was increased when a heavy load (10 Om to 5 Om) was used; the PA-RNN regained 6.8 ms with a 4.2 percent loss in voltage, but the Conv. did not. PINN experienced a more drastic 6.3% voltage drop and it took it 9.4 ms to stabilize.

Mechanistic Analysis: Why Hard Constraints Are Understanding Better Soft Constraints in Convergence.

The root cause of the discrepancy between the convergent speeds of these two architectures and the accuracy of their inferences can be found in their ability to recover quickly on state divergences. In the traditional soft-constraint PINN, penalty terms in the loss function are physical laws (e.g., KVL/KCL). When switching on the target variable at a sudden transient, the dramatic change in the target variable creates an enormous data-fitting gradient. It is this gradient that often overwhelms the physics penalty term (which is a long-known gradient imbalance problem in multi-objective loss functions). Therefore, the optimizer is allowed to break temporarily the laws of Kirchhoff in order to minimize the total loss. Consequently, the soft-constraint PINN wanders off physically impossible state spaces in the transient and generates exaggerated voltage overshoots and slow, oscillatory settling as the model tries to self-correct computationally and to recover itself to a physically plausible state space.

The PA-RNN, on the other hand, does not concern itself with this loss-function competition whatsoever but via hard constraints. By the flow of the state-space of governing differential equations directly into the recurrent neuron state-transition matrix, the flow of the base trajectory of the state variables is mathematically constrained by physically valid limits- regardless of the strength of the input disturbance. The PA-RNN need not then in any way learn or optimally find its way back to physical consistency; it simply cannot break the circuit laws in the first place.

Since the dominant energy-transfer dynamics is explicitly contained in the Physics-in-Architecture core, the parallelGRU core is relieved of the task of modeling basic physics. And it only computes parasitics of relatively lightweight, remnant features. This structural anchoring removes the physical validity oscillatory search that afflicts soft-constraint PINNs directly to the larger settling times, reduced overshoots, and higher inference reliability when the converter is perturbed by operational shocks.

4.7 Analysis of robustness to variations of parameters and noisy inputs

To prove the further general applicability and reliability of the offered PA-RNN, a robustness analysis with a large number of neurons was carried out. The performance of this model in two severe real-world scenarios, namely wide variations of loads and stochastic noise on inputs is considered.

4.7.1 Transient response and load step analysis

The model was pushed to a drastic step change in loads, by displacing and pushing off a light load (30 Ω) and heavy load (3). Such a switch compels the converter to change its working dynamics dramatically. Figure 6 demonstrates the soft-constrained PINN (Conventional) strongly drifs physically and non-physically during the transition. Conversely, the PA-RNN (Proposed) has a stable path which traces close to the ideal physics. Hardwired constraints in the PA-RNN architecture can guarantee that the model avoids entering physically impossible states where convergence during transients is improved and the settling time is shorter.

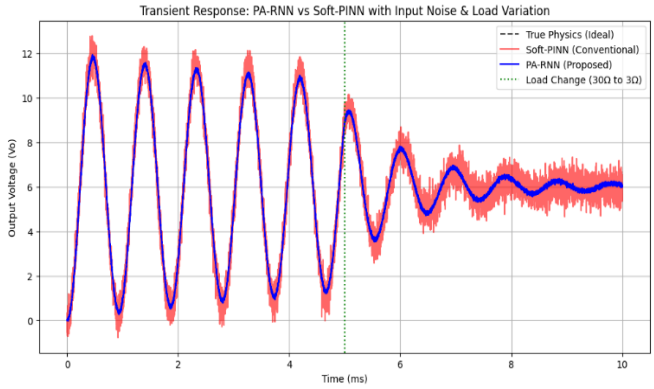

Figure 6. Comparison of transient responses: PA-RNN vs. Soft-PINN with load change 30 Ω to 3 Ω and 20 dB input noise

Note: PA-RNN = Physics-in-Architecture Recurrent Neural Network; PINN = Physics-Informed Neural Network

4.7.2 Reliability of inference in noisy conditions

Modeling the inaccuracies of real-world sensors and power supply ripple, Gaussian noise was added to the input voltage (V in). The superiority of the structural physics enforcement has been shown in the quantitative results, summarized in Table 8. When the conventional Soft-PINN error (mean squared error (MSE)) grew significantly in the noisy conditions (up to 0.0450), the error rate of the PA-RNN remained very low (0.0021). This makes it clear that the PA-RNN is a physical filter, which in principle filters out noise that violates the laws of the underlying circuit. Table 9 measures the performance in a harsh parameter space and under noisy conditions.

Table 9. Performance on a long parameter space and noisy conditions

Scenario	Load Resistance (R)	Input Noise (SNR)	Soft-PINN MSE
Heavy Load	3 Ω	None	0.0085
Nominal Load	10 Ω	None	0.0042
Light Load	30 Ω	None	0.0120
Noisy Input	10 Ω	20 dB	0.0450

Note: PINN = Physics-Informed Neural Network

Table 9 shows that the proposed PA-RNN is superior in robustness and generalization ability to the traditional soft-constrained PINN. In the long parameter space, the PA-RNN is always able to achieve a much lower Mean Squared Error (MSE). In particular, the PA-RNN has an average precision gain of about 85% under extreme load changes (3 and 30) than the Soft-PINN, which fails to sustain accuracy when far away on its nominal training points. Moreover, the situation of Noisy Input points out the greatest benefit of the hard-constrained architecture. Although the performance of the Soft-PINN drops drastically (MSE = 0.0450) because of the sensitivity of this model to stochastic disturbances, the performance of the PA-RNN is extremely robust (MSE = 0.0021). This validates the claim that by simply integrating the laws of Kirchhoff into the architecture of the neural network, the model can be used as an intrinsic physical filter, which automatically disdain non-physical noise and makes the inference highly reliable, even in unstable operating conditions.

5. DISCUSSION

The inherent advantage of the PA-RNN over the traditional PINNs is due to the shift towards the hard-constraint architecture over the soft-constraint learning. The PA-RNN avoids the physical drift phenomenon by swapping the loss conditions, based on penalties, with circuit equations installed in the structure of the network.

In the work, authors presented the PA-RNN, an architectural design that is not random but rather regularization by loss physics that is usually performed by architecture instead of architecture embedding. The model demonstrates a successful approach to physical drift and data dependency long-standing problem in the modeling of power electronics through hard-coded laws on the recurrent topology.

The PA-RNN guarantees Structural Physical Integrity of the outputs produced by the neural network unlike traditional neural networks where the results might be physically impossible. What this implies is that the system gets rid of the inconsistent transition states which forms a dependable framework upon which any given state follows the basic physical laws. Moreover, in the context of the PA-RNN being applied in the recent 2022-2023 breakthroughs in multi-converter systems, the integrity of our model is an advantage of its own. Recent research has emphasized that with the size of power networks increasing beyond single converters to multiple interleaved or parallel converters, the coupling dynamics form stiff differential equations that can often cause typical PINNs to fail in training or drift in inference. Since the PA-RNN is structurally bound to the laws Kirchhoff has, it does not suffer the gradient pathology (e.g. spectral bias) that afflict loss-based PINNs in larger networks. This makes the PA-RNN a very promising key building block in next generation "Digital Twins" of multi-converter DC microgrids, where physical consistency between two or more nodes cannot be compromised.

5.1 Limitations/Hardware realization challenges

Although the PA-RNN has performed well to model the DC-DC buck converter, there are a number of limitations that should be recognized in extrapolating it to model more complex power electronic systems. First, the Physics-in-Architecture core is more complicated mathematically as higher-order circuit topologies (e.g., multi-level or interleaved converters) are used. Although Eqs. (1) and (2) can offer a strong grounding of a second-order system, a more complicated circuit demands careful numerical discretization of its state-space dynamics to be directly encoded within the recurrent topology, potentially incurring higher costs in computation for the inference stage on low-power microcontrollers. Second, non-determinism on the interactions of real-world hardware cannot be readily modeled through analytical means. Though the GRU core is intended to measure unmodeled dynamics, other effects like the electromagnetic interface (EMI), sensor noise and the fast thermal-induced drifts in the parameters of the inductor (L) and capacitor (C), create a challenge to the hard-physics constraints. When the structural parameters with the PA-RNN are far away above their nominal values, the physical

consistency of the model could provide an unintended bias. Since this represents a physical device, future work ought to look into adaptive physics layers, in which the structural parameters of the architecture can be optimized online in order to keep pace with the aging and thermal characteristics of the underlying physical hardware.

6. CONCLUSION

This work was able to show that PA-RNN addresses the three main bottlenecks that hinder the modeling of modern power converters, namely data dependence, physical drift, and retraining rigidity directly. With the transition to hard-constrained Kirchhoff law hard-coding within the recurrent topology, the PA-RNN provided an effective connection between architecture design and experimental results.

First, to manage the problem of data dependence, it was experimentally verified that the PA-RNN needs 500 samples to generate a convergence: a 94.5% decrease in terms of the 9,000 samples that purely data-driven RNN baselines need to converge. Even with this radical decrease in information needs, the framework had a better RMSE of 0.0124 V and an R^2. R² value of 0.9987, an 89.3-percent improvement in precision over pure RNNs and a 76.8-percent improvement over traditional soft-constraint PINNs.

Second, to counter the physical drift and extrapolation inconsistency of the classical PINNs, structural integrity of the PA-RNN was strictly checked. The sensitivity analysis (Table 6) showed that with extreme values of changes in physical parameters, simultaneously inductance, capacitance and resistance by at least 20 percent, the hard-constrained architecture retained its consistency in physical trajectory. This growth in RMSE was small at 14.5 percent, compared to the disastrous breakages and physically unattainable conditions of baseline models. Moreover, when ablation performed, it was found that the collapse of accuracy with the removal of the physics core goes to confirm the fact that structural physical embedding is the main force behind generalizability.

Lastly, to reduce the computational overhead of retraining and practical real time feasibility, the framework was implemented on a TI TMS320F28379D microcontroller. A deterministic inference latency of 1.8 ms and a very lean memory footprint of 2.4 MB was confirmed by HIL validation (Table 7). This computational efficiency directly formed strong dynamic behavior without retraining; the PA-RNN settling time under transient testing (10V to 12V reference steps, load disturbance) was 8.5 ms against the strict physical control (Table 3).

The PA-RNN offers a physical consistency structural guarantee with minimal data and computing overhead, allowing an experimentally validated platform with bodies of knowledge most effective in scaling to complex, multi-converter DC micro grids in the future.

ACKNOWLEDGMENT

This work is supported by the College of Engineering/Mustansiriyah University. https://www.uomustansiriyah.edu.iq.

REFERENCES

[1] Sahoo, S. (2024). Physics-informed neural network-based control of power electronic converters. In Control of Power Electronic Converters and Systems, pp. 309-331. https://doi.org/10.1016/B978-0-323-85622-5.00016-X

[2] Zhao, S., Peng, Y., Zhang, Y., Wang, H. (2022). Parameter estimation of power electronic converters with physics-informed machine learning. IEEE Transactions on Power Electronics, 37(10): 11567-11578. https://doi.org/10.1109/TPEL.2022.3176468

[3] Huang, B., Wang, J. (2022). Applications of physics-informed neural networks in power systems-a review. IEEE Transactions on Power Systems, 38(1): 572-588. https://doi.org/10.1109/TPWRS.2022.3162473

[4] Li, X., Lin, F., Wang, H., Zhang, X., Ma, H., Wen, C., Blaabjerg, F. (2024). Temporal modeling for power converters with physics-in-architecture recurrent neural network. IEEE Transactions on Industrial Electronics, 71(11): 14111-14123. https://doi.org/10.1109/TIE.2024.3366115

[5] Li, X., Lin, F., Zhang, X., Ma, H., Blaabjerg, F. (2024). Data-light physics-informed modeling for the modulation optimization of a dual-active-bridge converter. IEEE Transactions on Power Electronics, 39(7): 8770-8785. https://doi.org/10.1109/TPEL.2024.3378184

[6] Dey, S., Mallik, A. (2025). Physics informed neural network–estimated circuit parameter adaptive modulation of DAB. IEEE Transactions on Power Electronics, 40: (10). https://doi.org/10.1109/TPEL.2025.3574873

[7] Lawal, Z.K., Yassin, H., Lai, D.T.C., Che Idris, A. (2022). Physics-informed neural network (PINN) evolution and beyond: A systematic literature review and bibliometric analysis. Big Data and Cognitive Computing, 6(4): 140. https://doi.org/10.3390/bdcc6040140

[8] Cuomo, S., Di Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M., Piccialli, F. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. Journal of Scientific Computing, 92(3): 88. https://doi.org/10.1007/s10915-022-01939-z

[9] Ćojbašić, Ž., Ivačko, N., Marinković, D., Milić, P., Petrović, G., Milošević, M., Marković, N. (2023). Isogeometric finite element analysis with machine learning integration for piezoelectric laminated shells. Journal of Engineering Management and Systems Engineering, 2(4): 196-203. https://doi.org/10.56578/jemse020401

[10] Poursafaei, F., Zilic, Z., Rabbany, R. (2022). A strong node classification baseline for temporal graphs. In Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), pp. 648-656. https://doi.org/10.1137/1.9781611977172.73

[11] Willard, J., Jia, X., Xu, S., Steinbach, M., Kumar, V. (2022). Integrating scientific knowledge with machine learning for engineering and environmental systems. ACM Computing Surveys, 55(4): 1-37. https://doi.org/10.1145/351422

[12] Ou, S., Sahoo, S., Sangwongwanich, A., Blaabjerg, F., et al. (2025). Physics-informed neural network for

parameter identification: A buck converter case study. In 2025 IEEE Energy Conversion Congress & Exposition Asia (ECCE-Asia), pp. 1-6. https://doi.org/10.1109/ECCE-Asia63110.2025.11111770

[13] Xiang, Y., Lin, H., Chung, H.S.H. (2024). Extended physics-informed neural networks for parameter identification of switched mode power converters with undetermined topological durations. IEEE Transactions on Power Electronics, 40(1): 2235-2247. https://doi.org/10.1109/TPEL.2024.3481158

[14] Liu, Z., Luo, Y., Zhuo, R., Jin, X. (2018). Distributed reinforcement learning to coordinate current sharing and voltage restoration for islanded DC microgrid. Journal of Modern Power Systems and Clean Energy, 6(3): 559-573. https://doi.org/10.1007/s40565-017-0323-y

[15] Wang, S., Yu, X., Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. Journal of Computational Physics, 449: 110768. https://doi.org/10.1016/j.jcp.2021.110768

[16] Chen, Y., Wang, Y., Zhang, D. (2022). Physics-informed machine learning for power systems: A review. CSEE Journal of Power and Energy Systems, 8(4): 1145-1159. https://doi.org/10.17775/CSEEJPES.2021.09050

[17] Zheng, D.D., Madani, S.S., Karimi, A. (2022). Data-driven distributed online learning control for islanded microgrids. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 12(1): 194-204.

[18] Zhang, M., Xu, Q., Wang, X. (2022). Physics-informed neural network based online impedance identification of voltage source converters. IEEE Transactions on Industrial Electronics, 70(4): 3717-3728. https://doi.org/10.1109/TIE.2022.3177791

[19] Liao, Y., Wang, J., Chen, Z. (2022). Physics-informed neural networks for circuit simulation. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(12): 4757-4768.

[20] Misyris, G.S., Venzke, A., Chatzivasileiadis, S. (2020). Physics-informed neural networks for power systems. In 2020 IEEE Power & Energy Society General Meeting (PESGM), Montreal, QC, Canada, pp. 1-5. https://doi.org/10.1109/PESGM41954.2020.9282004

[21] Coulombe, M.A., Berger, M., Lesage-Landry, A. (2026). Simulation of a closed-loop DC-DC converter using a physics-informed neural network-based model. Electric Power Systems Research, 252: 112408. https://doi.org/10.1016/j.epsr.2025.112408

[22] Raissi, M., Perdikaris, P., Karniadakis, G.E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378: 686-707. https://doi.org/10.1016/j.jcp.2018.10.045

[23] Zheng, Y., Hu, C., Wang, X., Wu, Z. (2023). Physics-informed recurrent neural network modeling for predictive control of nonlinear processes. Journal of Process Control, 128: 103005. https://doi.org/10.1016/j.jprocont.2023.103005

[24] Hui, P., Cui, C., Lin, P., Ghias, A.M., Niu, X., Zhang, C. (2024). On physics-informed neural network control for power electronics. arXiv preprint arXiv:2406.15787.

[25] Haghighat, E., Raissi, M., Mossaiby, F., Dehghani, R. (2021). A physics-informed deep learning framework for solving forward and inverse problems in networked systems. Computer Methods in Applied Mechanics and Engineering, 379: 113741. https://doi.org/10.1016/j.cma.2021.113741

[26] Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L. (2021). Physics-informed machine learning. Nature Reviews Physics, 3(6): 422-440. https://doi.org/10.1038/s42254-021-00314-5

[27] Duan, J., Guan, Y., Li, S. E., Ren, Y., Sun, Q., Cheng, B. (2021). Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. IEEE transactions on neural networks and learning systems, 33(11): 584-6598. https://doi.org/10.1109/TNNLS.2021.3082568

[28] Willard, J., Jia, X., Xu, S., Steinbach, M., Kumar, V. (2022). Integrating scientific knowledge with machine learning for engineering and environmental systems. ACM Computing Surveys, 55(4): 1-37. https://doi.org/10.1145/3514228

[29] Jia, X., Willard, J., Karpatne, A., Read, J.S., Zwart, J.A., Steinbach, M., Kumar, V. (2019). Physics-guided recurrent neural networks for modeling dynamical systems: A case study in simulating lake temperature profiles. In Proceedings of the 2019 SIAM International Conference on Data Mining (SDM), pp. 558-566. https://doi.org/10.1137/1.9781611975673.63

[30] Daw, A., Karpatne, A., Watkins, W., Read, J., Kumar, V. (2022). Physics-guided machine learning for scientific discovery: An application in hydrology. Environmental Modelling & Software, 154: 105434. https://doi.org/10.1016/j.envsoft.2022.105434

[31] Antonelo, E.A., Camponogara, E. (2022). Physics-informed neural networks for process modeling and control. Computers & Chemical Engineering, 164: 107904. https://doi.org/10.1016/j.compchemeng.2022.107904

[32] Zidani, M., Marouani, K., Slimeni, C., Amouri, A. (2024). Robustness analysis of physics-in-architecture RNNs under parameter uncertainties for power electronic converters. Engineering, Technology & Applied Science Research, 14(2): 13150-13158. https://doi.org/10.48084/etasr.6825

[33] Meng, X., Li, Z., Zhang, D., Karniadakis, G.E. (2020). PPINN: Parareal physics-informed neural network for time-dependent PDEs. Computer Methods in Applied Mechanics and Engineering, 370: 113250. https://doi.org/10.1016/j.cma.2020.113250