Neural Computing and Applications (2024) 36:20807–20822
https://doi.org/10.1007/s00521-024-10293-0

ORIGINAL ARTICLE

Circuit topology aware GNN-based multi-variable model for DC-DC converters dynamics prediction in CCM and DCM

Ahmed K. Khamis[1,2] · Mohammed Agamy[1]

Received: 26 February 2024 / Accepted: 29 July 2024 / Published online: 16 August 2024

Abstract
A regression model based on graph neural network, tailored for electric circuit dynamics prediction is introduced, providing converter performance predictions on converter circuit level and internal parameter variations. Regardless of the number of components or connections present in a converter circuit, the proposed model can be readily scaled to incorporate different converter circuit topologies. Moreover, the model can be used to analyse converter circuits with any number of circuit components and any control parameters variation. To enable the use of machine learning methods and applications, all physical and switching circuit properties such as converter circuits operating in continuous conduction mode or discontinuous conduction mode are accurately mapped to graph representation. Three of the most common converters (Buck, Boost, and Buck-boost) are used as example circuits applied to model and the target is to predict the gain and current ripples in inductor. The model achieves 99.51% on the R^2 measure and a mean square error of 0.0263.

Keywords Power electronics · Bond graph · Graph neural networks (GNN) · Machine learning

1 Introduction

Artificial intelligence has been incorporated as deep learning (DL) models in a wide range of disciplines. In particular, the use of recurrent neural networks (RNN) for sequential processing and convolutional neural networks (CNN) in electrical and renewable energy applications has been gaining momentum [1, 2]. Recently, graph neural networks (GNN), which model patterns in graph-structured data, have seen a surge in popularity. These networks are particularly advantageous for representing electrical circuit structure, as graphs are a natural data format for expressing such information.

In [3], GNNs were proposed as suitable alternatives to shallow methods or mathematical optimization techniques for circuit optimization/classification needs and multiple applications (e.g., transistor sizing, capacitor value optimization). [4, 5] used a reinforcement learning (RL) agent to select optimal parameters via rewarding based on figure of merit (FOM) when circuits were represented as graphs (nodes/edges refer to components/wires, each transistor embedded with a vector). [6] used differential neural network (DNN) for mapping a circuit to its corresponding transfer function, but applicable only for a specific topology. [7] combined feature maps of nodes via GNN to simulate a distributed circuit's electromagnetic properties. [8] used DeepGEN for predicting ladder and two-stage operational amplifier circuits with up to 10 branches, but lacked description of connection type and other elements, e.g., frequency, phase shift. [9] used GNN to identify symmetry constraints in analog circuits and proposed extending it to other constraints. [10] represented elements as heterogeneous multi-graphs and set four types of edges.

GNNs are not only capable of quickly training on graphs, but also generalizing to large datasets, and learning order permutation invariant representations from the graph modelling approaches, but they have also been applied to circuit design [7, 11, 12], though structure-based predictions in switching converter circuits have yet to be

✉ Ahmed K. Khamis
akhamis@albany.edu

Mohammed Agamy
magamy@albany.edu

[1] ECE Dept., University at Albany SUNY, Albany, NY, USA

[2] EE Dept., Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt

addressed. This paper proposes a framework that utilizes the use of graph representing circuits and GNNs to derive predictions of converter internal states while considering the control variables applied to converter like frequency and duty cycle as well as circuit component variations, with test cases of different and complex DC-DC converter circuits. The paper is divided into **six** sections, where section II shows the advantages of the proposed framework and its utilization for converter states predictions. Section III briefly discusses the circuit representation technique utilized in this framework and how to integrate it with GNNs, which was previously described and illustrated in details in [13]. Section IV shows the mathematical and logical development and utilization of the proposed regression model as well as its capability to handle different circuit complexities in terms of computational and space complexities. Section V expands the problem into a multi-variable regression problem and introduces multiple DC-DC converter internal states estimates, showing results of supervised and unsupervised learning. Finally, the model performance including error and accuracy for circuit and control parameter variations are analysed.

2 Advantages of ML framework in converter modelling

Power electronic converters are traditionally modelled using analytical differential equations that describe their steady-state and dynamic behavior. However, these equations may not capture the complexity and nonlinearity of the converters accurately, and may require simplifying assumptions and approximations. Machine learning (ML)-based models offer an alternative approach that leverages data from simulations or experiments to learn the input–output relationships of the converters. ML-based models have several advantages over classical models, such as:

- High-dimensional and noisy data handling without losing accuracy or generalization ability [14].
- Operating conditions and environmental variables adaptations [15].
- Very complex and time consuming task implementations like optimal design, control, fault diagnosis, efficiency and reliability assessment and of the converters design by using data-driven optimization techniques and feature extraction methods [16, 17].

Therefore, ML-based models have the potential to create digital twins of power electronic converters that can support their analysis, optimization, and maintenance in practical applications. Contributions included in this paper are listed as follows:

- A regression model is introduced, tying the electric circuit structure with circuit performance regardless of number of circuit components or the layout of the circuit. Unlike other machine learning regression models that rely on hyperparameter tuning of hidden layers, the proposed model leverages the graph representation to learn the relationship between the circuit structure and its performance metrics, such as current ripples and voltage gain.
- The proposed model can estimate converter performance based on circuit parameters, including circuit elements variations and controller signal variations and for different converter circuit topology variants. The model can handle different types of converters, such as buck, boost, and buck-boost, and differentiate between operating modes such as continuous conduction mode (CCM) and discontinuous conduction mode (DCM).
- The proposed model can differentiate and identify different circuit structures even if they have the same number of components by unsupervised means. The model can cluster different circuit topologies based on their graph features and performance metrics, without requiring any prior knowledge or labels.
- The proposed model accommodates different circuit and control parameters, for which the model's accuracy is tested. Histograms are used to evaluate the model's accuracy on three widely used converters (buck, boost, and buck-boost) under different operating modes (CCM and DCM), versus the circuit elements variations (inductance and resistance) and control variables (frequency and duty cycle) independently.
- The paper presents a case study of a multi-variable regression problem that illustrates how the model can be used to analyse the performance of different converters under various scenarios. The paper also shows how the model can scale to larger and more complex circuits by using graph neural networks (GNNs) to learn from the graph representation.

3 Circuit representation in ML domain

This work proposes converter dynamics predictions based on the physical connection and operating circumstances of a converter, based on circuits to machine learning (ML) domain mapping approach published in [13, 18–20]. A comparison between different graphical circuit representation techniques is given in Table 1, while a review of the literature on circuit representation as graphs in previous research studies is shown in Table 2. Bond graph modelling with switching circuit representation is used to transform circuits to graphs, from which a dataset is created. These

Table 1 Comparison between different circuit representation techniques

Method	Representation methods	Merits	Drawbacks
Graph theory	Component terminals are nodes. Circuit elements are edges	Multi-discipline physics-based modelling technique. More intuitive graph for human reader	Converter modelling foundations (duty cycle, CCM and DCM, etc.) are missing/never been addressed No research on graph identifiability from graph to circuit. Circuit graph can be defined using three matrices as shown in [21]
Bond Graph	Elements and connections are nodes with different attributes	Solid foundations on circuits/converter modelling in CCM and DCM. BG is a linear transformation and is mathematically identifiable as shown in [22]. Multi-discipline physics-based modelling technique. Generated graph can be defined with one Adjacency matrix. Maintains causality invariance of the system for any operational mode, i.e., the state vector resulting from state equation of the system does not change for any operating mode	Non-intuitive modelling technique. Added complexity of causality assignment. Can yield a bigger graph than graph theory method
Y admittance matrix	Circuit buses are nodes. Connections between buses are edges	Easy and well-known methodology for circuit representation	Used only for power system representation. Node count is independent from number of components. Number of circuit sources can't be extracted. System components can be lumped altogether and information about element count is lost. Never been used in converter modelling

datasets then undergo feature extraction and a graph neural network (GNN) model is applied with regression to obtain circuit predictions for unseen circuits. Additionally, circuit representation characteristics such as permutation invariance and scalability, as well as aspects tying circuit structure/behavior to corresponding graphs are discussed. A detailed and complete comparative study between graphical circuit representation is given in [13].

3.1 Converter circuits bond graph representation

An electrical circuit is composed of five main components that influence the electric current: resistors, inductors, capacitors, voltage source, and current source. These components can be modelled and analysed using bond graph (BG) elements and their mathematical relations, which is a graphical tool that depicts the energy flow and power exchange in a system. A bond graph consists of circuit elements that represent different forms of energy storage, dissipation, conversion, and sources, i.e., voltage and current sources, inductances, capacitances and resistors. These elements are to be connected to junction elements, which represent the circuit connection. Junction elements include zero and one junctions. Circuit elements are assigned to zero-junctions and one-junctions, following Kirchhoff's voltage law (KVL) and Kirchhoff's current law (KCL). In other words, circuit elements are assigned to zero junctions when connected in parallel and where the voltage is equal for all connected elements, while are being assigned to one-junctions when connected in series and where the current is equal for all connected elements. Comparable to the continuous circuits discussed in [13, 18, 19], the switching circuits utilize switching cells. Bond graph (BG) formulation [27] is used to convert converter circuit topologies into graphs, which incorporate switching cells modelled as 1 s- and 0 s-connections governed by its control variable D (analogous to duty cycle in circuits analogy). As depicted in Fig. 1a, the Single Pole Double Throw switching cell can be reduced to two Single Pole Single Throw cells (SPST). Every SPST is represented as a 1 s-junction with two flow determining bonds. The physical realization is complete when the current interruption, when the SPST switch is turned OFF, is depicted as one flow decider bond modelled as a zero value current source and the other flow decider bond still linked to the system. The control signals that control the junction flows are D and $\bar{D}$ (as in the physical duty cycle notion of the converter circuits). SPST switches combinations are modelled in bond graph notation as (0 s and 1 s) junctions, indicating that the system state may become discontinuous [28, 29]. Figure 2 shows the graph representation of different converters with embedded node and edge features.

Table 2 Review of circuit representation as graphs

References	Node features	Edge features	Circuit representation		Task	Network type
			Circuit components	Connections (Series/Parallel)		
[4]	DC operating points, One-hot encoding of simulation step, Transistor parameters, Internal capacitances	Featureless	Every circuit element is represented as node, where node features define the element type and DC operating conditions. No indication was given on connection representation, or its effect on analog circuit performance.		Learning design policy for selecting optimal circuit parameters.	RNN+RL
[5]	One-hot encoding of element type Circuit order, Passive and active characteristics	Featureless				GCN+RL
[23]	Gate logic level, Controllability, Observability	Featureless	Limited circuit representation in the form of connected nodes according to the physical connection.		Determine whether an observation point should be added on the output port or not	Meta-path + GCN
[6, 7]	Subcircuit coordinates, Center position of the Subcircuit, Angular position of the slit	Position of the two subcircuits, Gap length, shift	System level representation, where every subcircuit is represented as a node, while edges between two nodes represent distance between two subcircuits.		Electromagnetic outputs prediction based on resonators relative positions	GCN
[24]	Operation type Bitwidth	Signal information	System level representation, where every node represents a microbench operation, while edges represent signals.		Operation Delay Prediction for FPGA HLS	GraphSAGE
[8]	One-hot encoding of terminal type, Device parameters	Featureless	Edges, but component terminals are represented as nodes	No direct indication of connection	DC output voltage prediction	Deep-GEN
[25]	Gate poly length, number of fingers, number of fins, number of copies, length of resistor, Capacitors, number of copies, net N	Featureless	Nodes	No direct indication of connection	Net parasitics Predictions based on physical devices parameters	GraphSage, Relation GCN and Graph Attention Networks
[9]	One hot encoding (Device/Pin) Path based feature	Featureless	Nodes represent component terminals and pins. Components can have multiple nodes representing Pins. Pin/Components are distinguished by node features. Power/GND is represented as I/O nodes.	No direct indication of connection	Binary classification of layout symmetry	GCN
[10]	Node type, Geometry, layer	Featureless	Devices and circuit elements	No direct indication of connection	Binary Classification of layout symmetry	Gated Recurrent Unit-based GNN
[26]	Device type, Functional Module, Current mirror, Differential pair, Active load, Device dimension, Device location	Horizontal and vertical distance between pins Pin metal layer, Pin length, Pin type	Nodes with different types	No direct indication of connection	Prediction of IC placement impact on circuit performance	GAT + Pooling (PEA)

Table 2 (continued)

References	Node features	Edge features	Circuit representation		Task	Network type
			Circuit components	Connections (Series/Parallel)		
Proposed [13]	Element ID, Normalized Component Values	One for continuous Circuits, Duty Cycle for switching circuits	Nodes with different types	One and zero nodes for every branch/voltage node	Different circuit topologies-based ML tasks (Classifier, Regression, Clustering)	GCN + Pooling

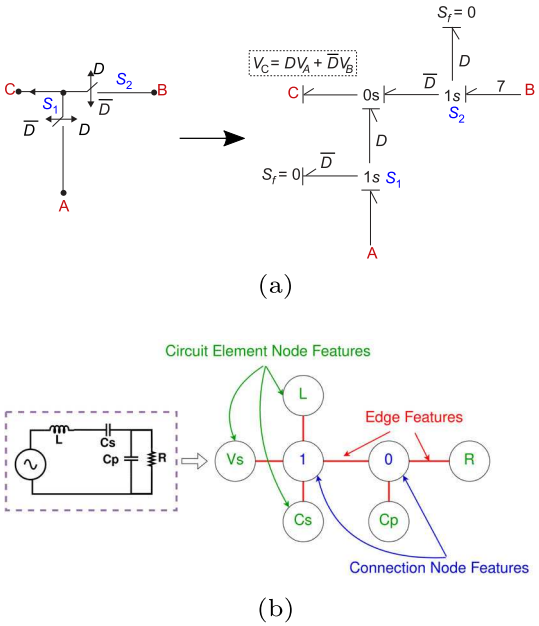


Fig. 1 **a** SPDT Switching cell represented as bond graph, **b** An LCC resonant circuit and its equivalent bond graph with node and edge features assignment

3.2 Circuits graph representation

The next step is to transform the BG formulation into a graph representation that contains all the information collected and simulated from the circuit, such as circuit types, classes, nodes, edges, and node and edge features. A graph representation of a continuous circuit based on the BG formulation, with some minor modifications for Switching circuits, is shown in Fig. 1b. Circuit elements and zero and one junctions are represented by nodes. Circuit connections between nodes are represented by edges. Node and edge features describe the operating condition of the circuit. In continuous circuits, edge features are set to one to indicate 100% connection between the corresponding nodes. The same notation is used for switching circuits. Node features indicate the type and value of the element in the circuit. Some properties of switching circuits need special attention like the duty cycle, which is a property of every switching circuit that physically indicates the percentage of time that the connection exists within a switching cycle. The duty cycle is mapped as a feature of the edges that connect to switching nodes (0 s and 1 s nodes). On the other hand, switching Frequency is a property of the switching cell, which is represented by one/zero switching junctions that are connected to a current source with zero value. This current source acts as a control source for every switch and interrupts the switch current with a frequency equal to the switching frequency. Therefore, including the switching frequency as a property of the BG control source is consistent with the physical properties of the circuit. More informative and in detail explanation of circuit to graph representation can be found in [13].

3.3 Graph neural network

Graph neural networks (GNNs) are to be used for graph-structured data, with two main categories: spectral GNNs, which operate in the spectral domain of the graph (e.g., GCN [30], GAT [31]), and spatial GNNs (e.g., GraphSAGE [32], R-GCN [33]). GCNs have advantages over traditional convolutional neural networks (CNNs) due to their capability to capture complex relationships between nodes in a graph and achieve better prediction accuracy [30], generalize to unseen data, and accommodate nodes of varying degree of connectivity, making them suitable for non-uniform data.

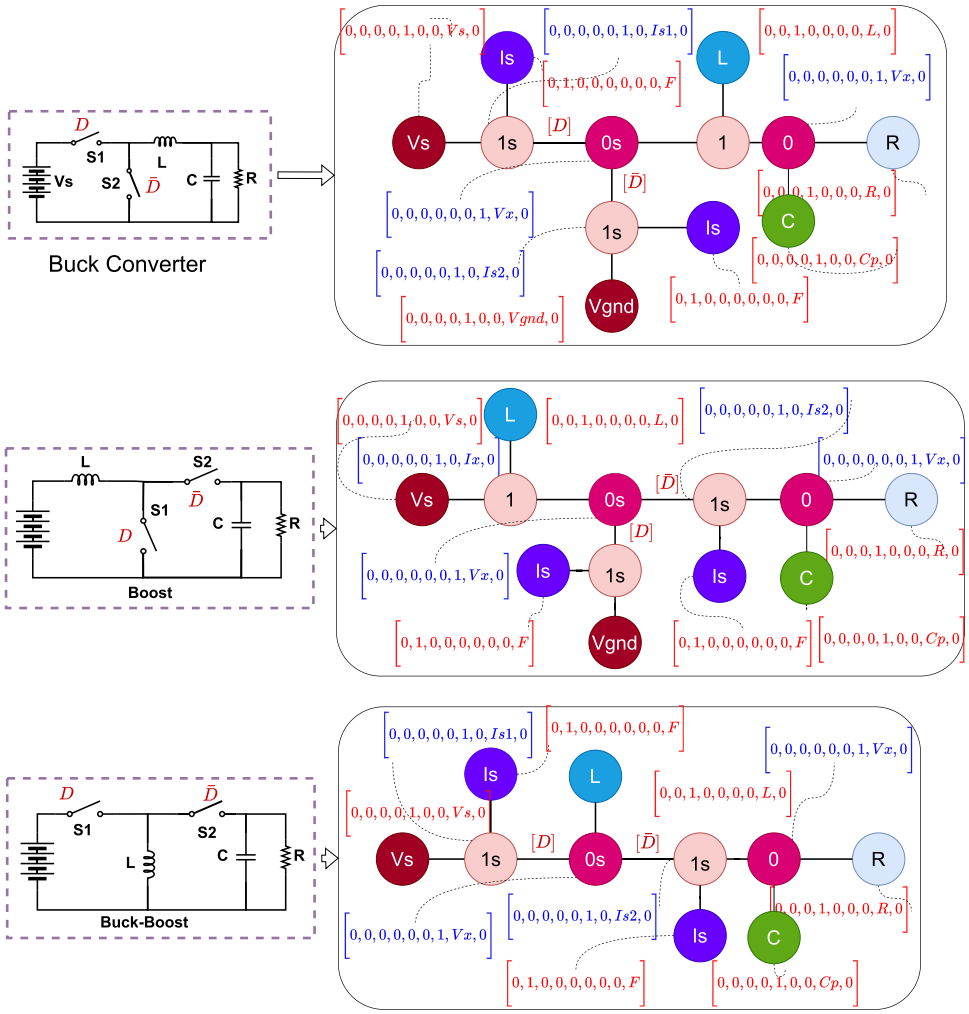


Fig. 2 Circuit diagrams and their equivalent graph with node features for Buck converter, Boost converter and Buck-boost converter

4 Regression model definition

Regression is the process of computing/predicting the output of a system given its input and parameters, which is also known as the forward problem. Mathematically, the system is a function f that maps an input space $\mathcal{X}$ to an output space $\mathcal{Y}$. The relation between the input $x \in X$ and the output $y \in Y$ is described by the dependent probability distribution P in Eq. (1) on $X \times Y$. Practically, the distribution P shown in Eq. (2) is known only through data samples z, which are the circuit simulation data obtained at different operating points, and are used as training set, which were simulated independently and within identically distributed data points.

$$P(x, y) = f(x)P(y \mid x) \tag{1}$$

$$\mathbf{z} = (\mathbf{x}, \mathbf{y}) = ((x_1, y_1), \ldots, (x_\ell, y_\ell)) \tag{2}$$

$$g(x) = \int_Y y dP(y \mid x) \tag{3}$$

The regression function can be mathematically defined in Eq. (3), which is a well-posed problem and has a unique solution that depends continuously on the input and parameters.

4.1 Regression model development

The goal of the proposed regression model is to obtain an approximation of f such that the approximated function ($\hat{f}$) can generalize well to new unseen data. The following assumptions are considered:

- A hypothesis space $\mathcal{H}$ which is defined as the set of all possible functions that can be used to model functions from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$.
- Probability distribution P on $\mathcal{X} \times \mathcal{Y}$.

The goal of the estimator is to find a function $f \in \mathcal{H}$ that minimizes the expected risk, which is a measure of the performance of a learning algorithm on new, unseen data because it quantifies the average loss incurred by the algorithm when making predictions over the entire joint

distribution of inputs and outputs. The expected risk defined by Eq. (4) is proportional to the mean squared error defined by loss function in Eq. (5):

$$I(f) = \int_{\mathcal{X}\times\mathcal{Y}} L(f(x), y) dP(x, y) \tag{4}$$

$$L(f(x), y) = (f(x) - y)^2 \tag{5}$$

Since the regression function in 3 is well defined and is the minimizer of the expected risk over the space of all the measurable real functions on X, g can be taken for an ideal estimator of the distribution probability P. However, due to the limited, finite, and possibly small number of simulation data points recorded ($\mathbf{z}$), the regression function cannot be perfectly constructed, and to overcome this problem, the solution to the regularized least squares problem is to be redefined as the estimator $f_{\mathbf{z}}$, which is indicated in Eq. 6 where Ω is a penalty term and ℓ is the number of regression variables.

$$f_{\mathbf{z}} = \min_{f\in\mathcal{H}} \left\{ \frac{1}{\ell} \sum_{i=1}^{\ell} (\Omega(f(x_i)) - y_i)^2 \right\} \tag{6}$$

Equation 6 are valid only for known dimension problems, but since the converter circuits are fed to the Regression model in non-Euclidean graph forms consisting of node and edge features, a transformation function (GCN) is used by applying feature transformation and aggregation operations to transform the graph to Euclidean space. The initial node features in (7) are initialized then fed to the graph processor, which are k layers of GCNs. Each node in the graph receives and sends messages from/to its neighbours and itself, which are weighted by their degrees and normalized by their square roots. The messages are aggregated by the sum and multiplied by a weight matrix, where an activation function is applied to obtain the new embedding for each node. The node wise convolution operation can be mathematically formulated as 9. where Θ^k is a weight matrix for the k-th neural network layer and σ is a nonlinear activation function, $\hat{A}$= A + I, where I is the identity matrix and $\hat{U}$ is the diagonal node degree matrix of $\hat{A}$. This allows the GCN to scale well, since the number of parameters in the model is not tied to the size of the graph. The node-wise formulation of feature update is given by Eq. (9), where $\hat{d}_i = 1 + \sum_{j\in\mathcal{N}(i)} e_{j,i}$ denotes the edge weight $e_{j,i}$ from source node j to target node i. Equation (10) defines the global pooling operation that averages the node embeddings into a single vector representation, denoted as V, where x_n is the embedding of node n, and N_i is the number of nodes in graph i.

$$X^{(0)} = E(X) \tag{7}$$

$$X^{(k+1)} = \sigma(\hat{U}^{-\frac{1}{2}}\hat{A}\hat{U}^{-\frac{1}{2}}X^k\Theta^k) \tag{8}$$

$$\mathbf{x}_i' = \Theta^\top \sum_{j\in\mathcal{N}(v)\cup\{i\}} \frac{e_{j,i}}{\sqrt{\hat{d}_j\hat{d}_i}} \mathbf{x}_j \tag{9}$$

$$\phi = \frac{1}{N_i} \sum_{n=1}^{N_i} \mathbf{x}_n \tag{10}$$

Predictions on graphs are obtained by applying linear transformation $V \mapsto \mathcal{R}$, which can be obtained by using linear functions with nonlinearity. The first linear layer can be interpreted as a linear transformation followed by a nonlinear transformation of the graph vector representation. Equation (11) defines the first fully connected layer, which takes the graph hidden vector representation ϕ as input and produces a nonlinear response $\mathbf{f}^{(1)}$ as output. The function $\sigma(\cdot)$ is the nonlinear activation function namely leaky ReLU, which is formulated in (12), hence controlling the slope of the negative part of the function by controlling the tunable factor α, while $w_0^{(1)}$ is the bias term of the first layer. Equation (13) defines the second linear layer, which takes the output of the first layer $\mathbf{f}^{(1)}$ as input and produces $f^{(2)}$ as output. The $f^{(2)}$ output is the prediction of the regression task that is comparable to the ground truth y. The estimator model flowchart is represented in Fig. 3.

$$f^{(1)}(\phi) = \sigma\left(w_0^{(1)} + \sum_{n=1}^{N} w_n^{(1)} \phi \right) \tag{11}$$

$$\sigma = max(\alpha\mathbf{input}, \mathbf{input}) \tag{12}$$

$$f^{(2)}\left(f^{(1)}(\phi)\right) = w_0^{(2)} + \sum_{i=1}^{U_1} w_i^{(2)} f_i^{(1)}(\phi) \tag{13}$$

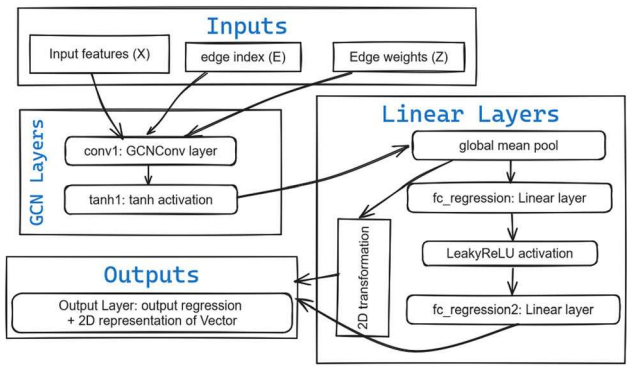


Fig. 3 Proposed model for circuit dynamics prediction

4.2 Model regularization

The dropout layer is chosen as the penalty term in Eq. (6) which reduces the overfitting of the predicted output to the ground truth, by introducing some noise and variability into the prediction. The inclusion of a dropout layer makes the prediction more generalizable to unseen data. However, there is a possibility that the accuracy and precision of the prediction are affected by introducing some error and uncertainty into the prediction. The dropout function used in this model can be defined in (14), where w_i is the value of the i-th weight, and $\mathbb{I}(w_i \neq 0)$ is an indicator function that returns 1 if w_i is non-zero and 0 otherwise. The mask function $\mathcal{I}$ has a probability of p of being zero for each element, and a probability of $1-p$ of being one.

$$\Omega = w_i \cdot \mathbb{I}(w_i \neq 0) \tag{14}$$

4.3 Circuit complexity investigation

In order to quantify the model's ability to handle more prediction outputs and more complex circuit structures, model performance metrics are to be used as judgement factors. Some possible metrics are:

- Accuracy: Evaluation of the GNN model prediction percentage, by using one of the most common evaluation metrics like F1-score, accuracy, precision or R^2, which will be shown in the case study in the next section.
- Computational scalability: Computational effort evaluation of model's performance, number of parameters, or memory usage against the increase in graph nodes and feature sizes without compromising the accuracy or performance.

Given G as the circuit graph, e as the number of edges, N as the number of nodes, d as the latent vector length, OL as output layer size and d_{in} as the feature vector length, the computational effort can be broken down to time and space complexities and are calculated for three GCN layers as:

- Time complexity: $O(3(e + Nd_{in}^2) + (N + dN) + 2d \times d + d \times OL)$
- Space complexity: $O(N + e + Nd_{in} + 3d + OL)$

Figure 4 shows the time and space complexity of the proposed model. (O) is the order of magnitude which defines the complexity growth proportional to the graph input size and number of features assigned for every node. Different converters and their operating modes are highlighted in the figure showing the memory and processing requirements to handle each circuit graph.

5 Multi-variable regression problem

This section presents a case study of three DC-DC converters for obtaining predictions based on circuit topology and component values. The main prediction targets are voltage gain and current ripples, which are governed by equations in Table 3, with the potential to scale up to include many more variables. The dataset contains graph forms of circuit data, as well as information about the prediction targets, obtained from simulations. It is observed that various factors such as resistance, inductance, duty cycle, switching frequency, converter type and operation mode are influential to the output voltage gain and current ripples. Thus, the solution search space becomes ubiquitous, rendering traditional analytical solvers inefficient.

5.1 Proposed GNN-based prediction model

To address the mentioned issues, a neural network model is proposed, which takes converter circuits in graph forms (G), node features (X) expressing element type and element value, adjacency matrix (A), edge features (e) as inputs, and outputs the predicted variables (Y) with output vector size being the number of predicted variables (ℓ). The mathematical representation of the regression model and the propagation of graph features across layers are given by Eqs. (15–20). Mathematically, this initial embedding function is represented by Eq. (7). The aggregation layer has multiple Graph Convolution Networks (GCN) that performs multiple message passing leaps to collect information about neighbouring nodes and keeps updating the latent dimensional vector with dimension d, which is mathematically represented as in Eq. (8).

$$Y = \text{Regression}(X, A, e) \tag{15}$$

Where

$$X \in \mathbb{R}^{N \times d_{in}} \tag{16}$$

$$Y \in \mathbb{R}^{C \times 1} \tag{17}$$

$$GCN^{(k)} : \mathbb{R}^{N \times d_{in}} \mapsto \mathbb{R}^{N \times d} \tag{18}$$

$$k \in \{0, 1, .., k-1\} \quad GMR : \mathbb{R}^{N \times d} \mapsto \mathbb{R}^{1 \times d} \tag{19}$$

$$FC : \mathbb{R}^{1 \times d} \mapsto \mathbb{R}^{1 \times \ell} \tag{20}$$

5.2 Obtaining predictions from circuit structure

Figure 5 shows a paradigm of obtaining predictions from circuits using a regression model. The model utilizes three GCN layers to exchange messages across nodes. The output is fed to the global mean readout (GMR) layer, which

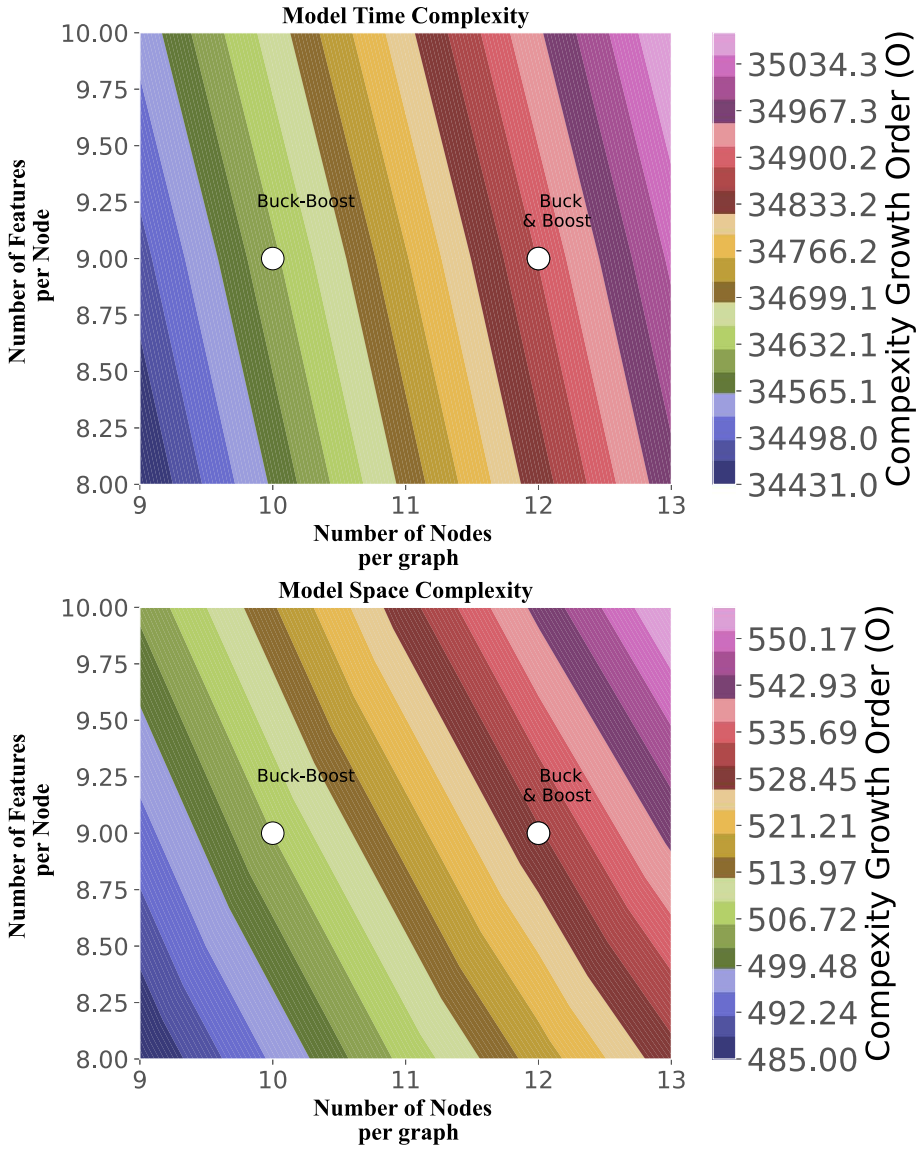
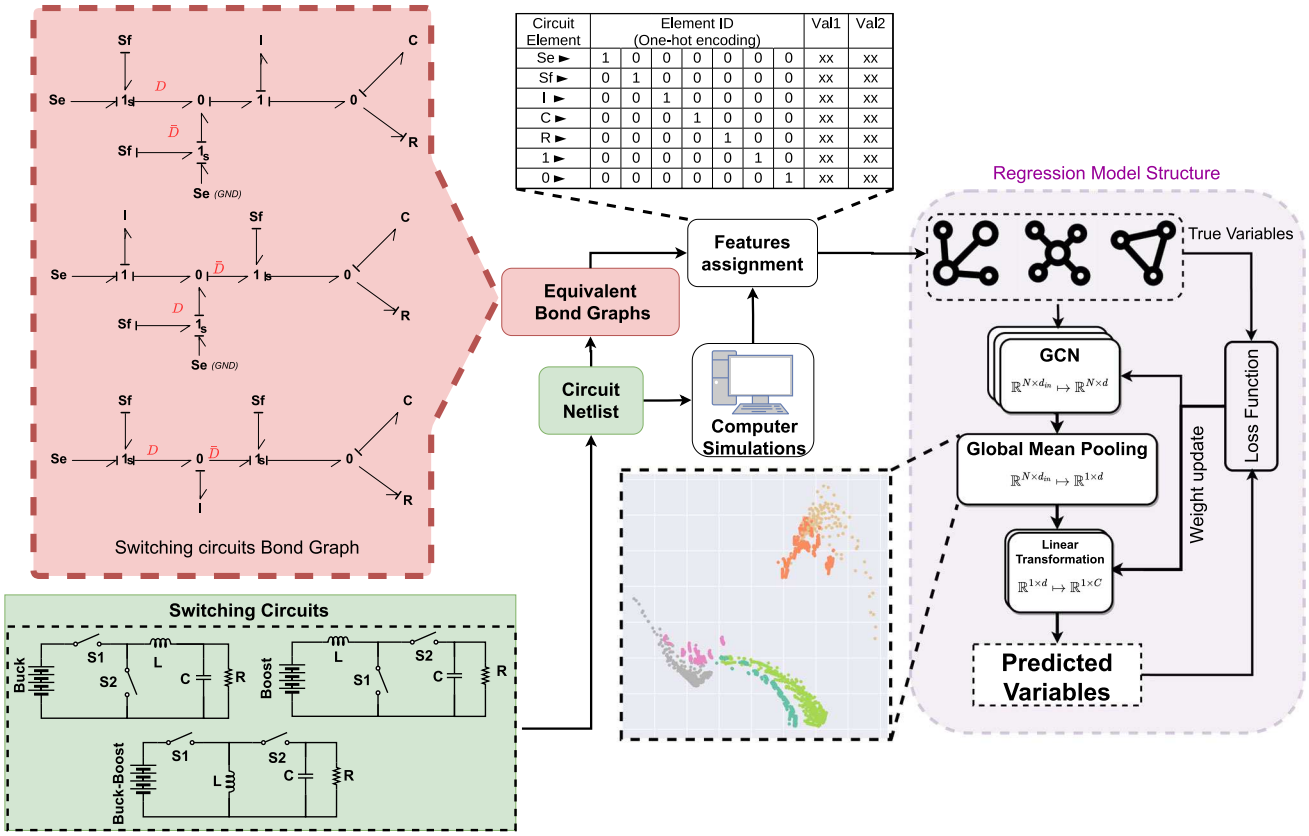


Fig. 4 Time and space computational complexity for each DC-DC converter

Table 3 Gain and CCM/DCM boundary formulas

	CCM voltage gain	DCM voltage gain	CCM/DCM boundary
Buck converter	D	$\frac{2D}{D+\sqrt{D^2+\frac{8L}{RT_s}}}$	$\frac{1}{2}(1-D_{crit})=\frac{L}{RT_s}$
Boost converter	$\frac{1}{1-D}$	$\frac{1}{2}\left(1+\sqrt{1+\frac{2D^2RT_s}{L}}\right)$	$\frac{1}{2}D_{crit}(1-D_{crit})^2=\frac{L}{RT_s}$
Buck/boost converter	$\frac{-D}{1-D}$	$D\sqrt{\frac{RT_s}{2L}}$	$\frac{1}{2}(1-D_{crit})^2=\frac{L}{RT_s}$

averages the processed node and edge features to an output dimension of d. The two fully connected (FC) linear layer is trained to linearly transform the averaged graph vector to desired output predictions by minimizing the mean square error loss function. According to the universal approximation theorem, neural networks with appropriate depth, FC layers may estimate any function without any limitation on the structure. Equations 15–9 express the regression model mathematics, while Eq. (11–13) are used to express the two-layer FC layer mathematically. Model is fed with a 18,000 graph dataset split 70% to 30% between training and testing datasets, and a separate unseen validation set of 2200 graphs. Datasets include captures of converter gain and inductor current ripples at various inductances, loads,



Circuit Element	Element ID (One-hot encoding)							Val1	Val2
Se ►	1	0	0	0	0	0	0	xx	xx
Sf ►	0	1	0	0	0	0	0	xx	xx
I ►	0	0	1	0	0	0	0	xx	xx
C ►	0	0	0	1	0	0	0	xx	xx
R ►	0	0	0	0	1	0	0	xx	xx
1 ►	0	0	0	0	0	1	0	xx	xx
0 ►	0	0	0	0	0	0	1	xx	xx

Fig. 5 Paradigm for obtaining predictions from circuits using regression model

frequency, duty cycle and converter topology in CCM and DCM. Training dataset is used in updating the linear layers and GCN layers weights through Adam optimizer running at 0.02 learning rate.

5.3 Dataset data range

The range of the datasets has to be properly identified, or else the resulting regression model may perform poorly as it will not be able to accurately extrapolate beyond the range of the dataset. Generally, selected algorithms should have the potential to accommodate minimum and maximum values in the datasets. Furthermore, the range of the datasets should effectively match the purpose of the machine learning model. It is essential for chosen algorithms to be capable of handling the full spectrum of values within the datasets. Moreover, the span of the datasets must align precisely with the intended function of the machine learning model. Should the scope of the datasets be excessively constrained, the model's capacity to adapt to parameter variations may be compromised, leading to less accurate output prediction. Conversely, an overly broad dataset scope can lead to a model that is overgeneralized, diminishing its precision due to erroneous predictions. The range of the utilized dataset including circuit and control parameters represented as node and edge features are listed in Table 4. The variables ranges [resistance (R), inductance (L), duty cycle (D) and frequency (F)] are selected so that the dataset include a wide range of CCM and DCM operation modes in the three converter classes.

5.4 Results analysis

5.4.1 Unsupervised learning clustering

The global mean pool layer outcome in Fig. 5 shows a 2D embedding, where each point represent a circuit graph. It is shown that the embedding contains the three converter topologies in each operating mode (CCM & DCM). The proposed model is capable of not only of differentiating the converter circuit topologies, but also recognize different operating modes (CCM & DCM). The unsupervised clustering of each converter topology occurs without prior knowledge of correlation between converter circuit graphs or the operation mode (CCM & DCM). Furthermore, the learnt graph embeddings represent the converter operating conditions (based on circuit and control parameters) per class basis. This proves that the regression model prediction takes into consideration the converter structure as well as operating conditions and can attain performance

Table 4 Dataset circuit parameters range

Parameter	Range	Notes
R	1 Ω↦ 20 Ω	Parameters based on equal CCM and DCM dataset of buck converter
L	1 μH ↦ 10 μH	
F	10 KHz ↦ 1 MHz	
D	0.01 ↦ 0.85	Limited to prevent infinite gain

prediction even with intricate converter dynamics as shown in Fig. 6.

5.4.2 Accuracy analysis

The R-squared which is a statistical measure of how much the model fits the data, is the percentage of the dependent variable variation that a model can explain. Equation 21 shows the mathematical formula for the coefficient of determination, where RSS is the sum of squares of residuals and TSS is the total sum of squares. It is also known as the coefficient of determination, and for the same data set, higher R-squared values represent smaller differences between the observed data and the fitted values. The score R^2 for the trained model is 99.49% when tested with the validation dataset, Two independent variables are being predicted in two operating modes: (• for CCM or △ for DCM) vs. simulated (line) natural log inductor current ripples and gain curves are shown for the three topologies. Obviously, even though the current ripples predictions vary with frequency, inductance, resistance, and duty cycle, the model is able to adapt all components and control variations. The second predictions are the converter gain shown in Fig. 7. In CCM, gain is dependent only on duty cycle. However, it nonlinearly depends on multiple factors like inductance, frequency duty cycle and resistor values in DCM. Despite the nonlinearity, the model is able to predict the gain and the current ripples utilizing the same training data. Notably, the model has learned it is the same converter working in two modes (CCM & DCM) and clustered each converter mode based on their topologies.

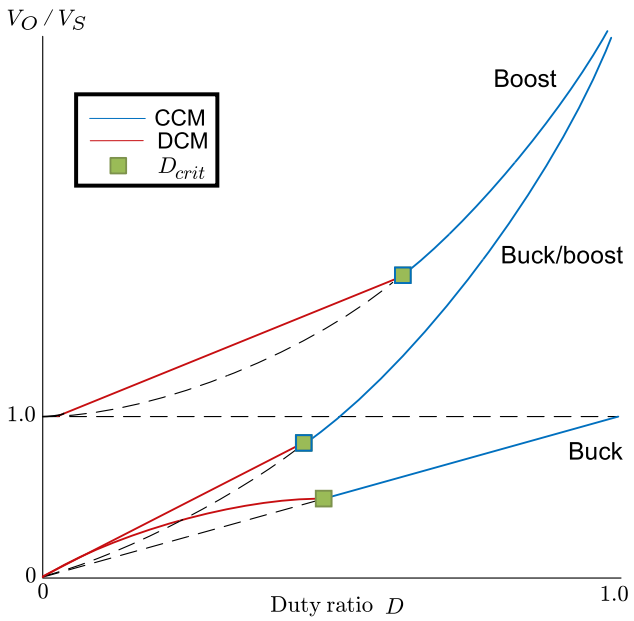


Fig. 6 Voltage gain curves of the three converters in CCM and DCM

$$R^2 = 1 - \frac{R_{SS}}{T_{SS}} = 1 - \frac{\sum_i (y_i - \hat{y_i})^2}{\sum_i (y_i - \bar{y_i})^2} \tag{21}$$

where, y_i = observed value, $\hat{y_i}$ = predicted value, $\bar{y_i}$ = mean of observed values.

6 Model performance analysis

The prediction error mean and standard deviation are crucial for its performance as these defining metrics highlight the accuracy of the model's predictions. A low error mean (μ) and standard deviation (σ) typically indicate a high-performing model, whereas a high error μ and σ indicate a low-performing model. Keeping both performance metrics in low values is necessary for optimal performance.

6.1 Circuit and control parameters sensitivity analysis

Parameter sensitivity analysis in regression models measures how changes in the parameter values affect the regression results. The main aim of performed parameter sensitivity analysis is to determine prediction error rate of the model in correspondence to different parameters variations. Two types of variable variation are performed from circuit prospective: the circuit parameters variation and control parameters variation. From a GNN prospective, these variations are node feature variations (resistance, frequency and inductance variations) and edge feature variations (duty cycle variation). It is important to show that the model corresponds to all circuit graph representation parameters and how effectively these variations are

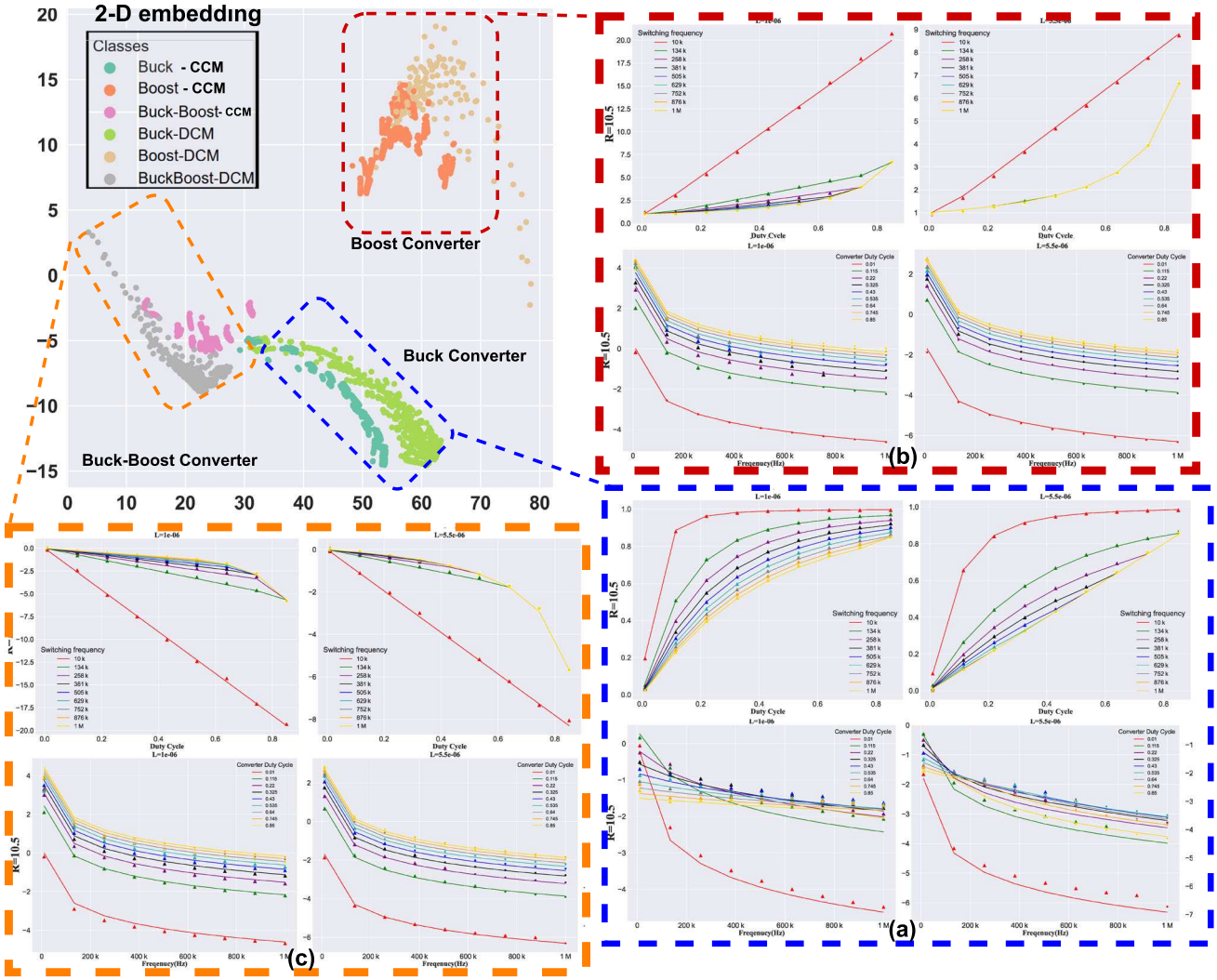


Fig. 7 2-D embeddings and their corresponding voltage gain and current ripples predictions based on converter type and operation mode for (a) Buck, (b) Boost and (c) Buck-boost converters. ●: CCM, △: DCM

reflected on the output. Figure 8a–c shows the actual vs predicted outputs of the model, namely, voltage gain and current ripples in CCM and DCM for the buck, boost and buck-boost converters, respectively. The straight line shows 100% accuracy prediction where $Y = \hat{Y}$. The proposed model remarkably predict both outputs with high precision and accuracy, scoring a minimum R^2 of 99.77% in predicting voltage gain and 98.63% in current ripples prediction. Figure 9 shows histograms of the prediction error distribution across the multi-variable regression model, while showing prediction error mean and standard deviation per converter class, circuit parameters and regression outputs in Table 5.

6.1.1 Circuit parameters variation (hardware variation)

Two types of variations are included in the proposed regression model, namely hardware and controller parameter variations. Hardware variation is when circuit component values are changed, which is a real life equivalent of adding more windings to the inductor or using another core type or adjusting the airgap.

Load variation: The histogram in Fig. 9a indicates error distribution in voltage gain prediction when exposed to changes in resistance across dataset range. The overall performance of the model was highly accurate, as indicated by the minimal prediction error, scoring the lowest in buck converter, with an error mean of zero when predicting the voltage gain. Furthermore, the error illustrated in Fig. 9b is minor in all converter classes when predicting current ripples, but it is more spread around error axis. On the other

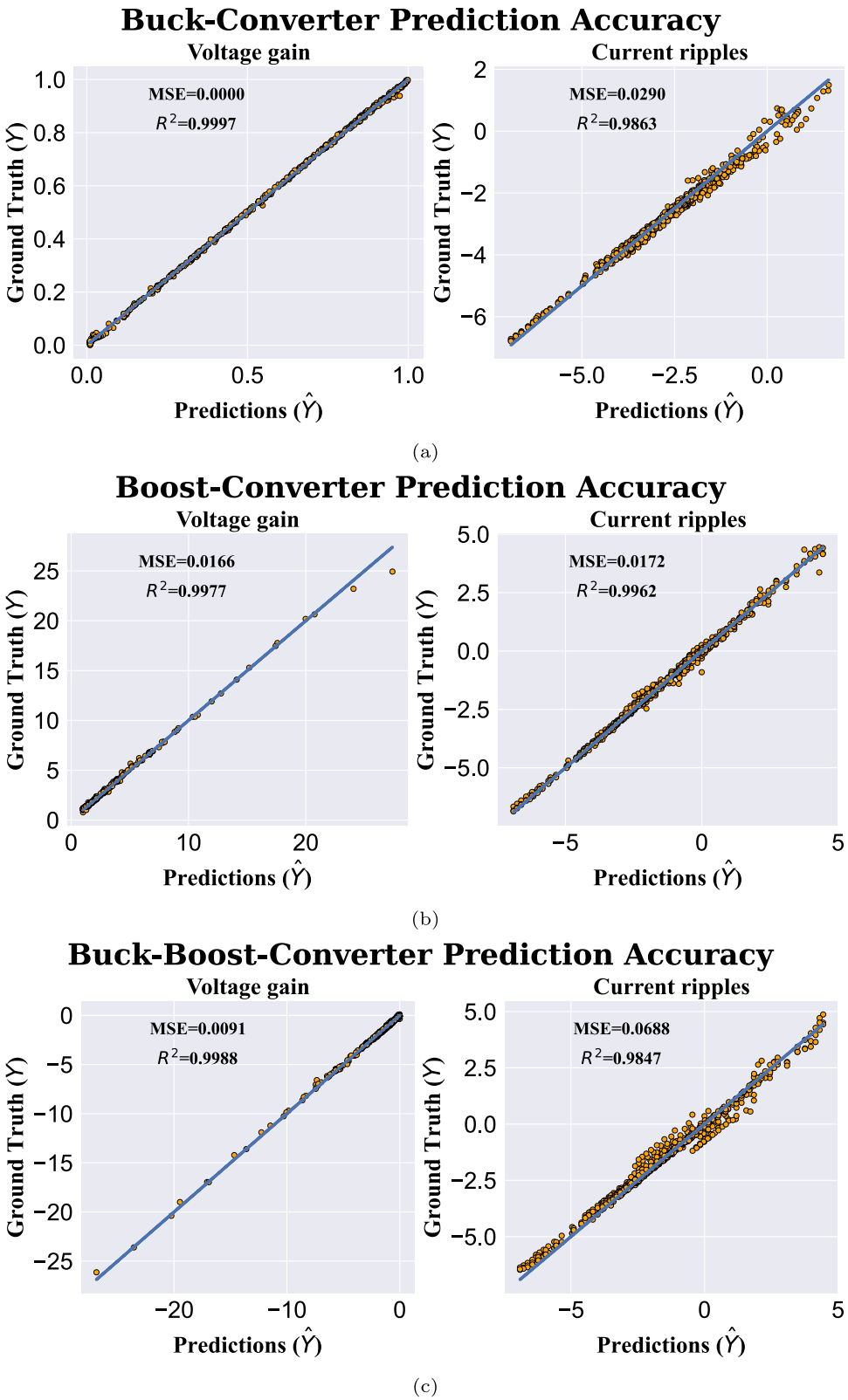


Fig. 8 Actual vs prediction in voltage gain and current ripples for three converter classes: **a** Buck converter, **b** Boost converter, **c** Buck-Boost converter

Fig. 9 Model absolute prediction error distributions when subjected to multiple variations

Table 5 Performance assessment for model output

Converter	Voltage gain		Current ripples	
	MSE	R^2	MSE	R^2
Buck converter	0	0.9997	0.0290	0.9863
Boost converter	0.0166	0.9977	0.172	0.9962
Buck-boost converter	0.0091	0.9988	0.0688	0.9847
Total MSE	0.0263			
Total $\mathbf{R}^2$	0.9951			

hand, the error mean of current ripples prediction of the boost converter was the lowest, while exhibiting an overall constant error mean across the resistance range in both output targets.

Inductance change: Figure 9c and d shows model error when subjected to changes in inductances on both output targets. The overall performance of the model is of high accuracy and prediction error is minimal. Predictions on buck converter are the most accurate and least error when predicting voltage gain. However, the boost converter is of the least error when predicting current ripples.

6.1.2 Control parameters variations

Controller parameters are changed like duty cycle and frequency, which are controlled by digital controllers running in real-time and require software changes. The proposed regression model can accurately predict the converter behaviour under these changes.

Switching frequency variation: Frequency is an important factor for converter operation mode, which is actively managed by a digital controller in real time, depicted as a node feature when represented in the circuit graph. In Fig. 9e and 9f, the variance in voltage gain prediction error is relatively low, with almost no changes across frequency range and converter class. On the other hand, when looking into current ripple prediction, the same pattern is observed, yet the prediction errors are comparatively higher, with a higher variance.

Duty cycle variation: Two major contributions are deduced from Fig. 9g and 9h which are the responsiveness of the model to duty cycle, which is represented as edge feature in circuit graph, and unchanged error mean when the duty cycle is varied from 0.01 and is limited to the practical value of 0.85 to prevent introducing infinite gain in the dataset. The voltage gain error is more significant in boost and buck-boost converters since gain vary significantly across duty cycle range and up to 25×.

7 Conclusion

The manuscript showed a framework for predicting DC-DC converter topologies performance including voltage gain and current ripples against variable control signal and component values. This framework is applicable to any circuit including continuous and switching circuits. By representing converter circuits as graphs models based on bond graph modelling technique, The developed GNN-based model was able to predict circuit performance information as well as identify the type of circuit even though it was not a planned task. This approach presents a very promising step towards AI understanding of circuit topologies and circuit analysis.

Data Availibility The data supporting the findings of this study are available from the corresponding author upon reasonable request. The data are not publicly available due to the following reasons: The data contain confidential information about the design and performance of the electric circuits, which are proprietary to the authors or their collaborators. The data are part of an ongoing research project that has not yet been completed or published. The data will be deposited in a suitable repository after the paper is accepted for publication and within reasonable time.

Declarations

Conflict of interest The authors declare that they have no financial or non-financial conflict of interest related to this work. The authors have not received any funding or support from any organization or entity that could affect the design, execution, analysis, or interpretation of the study. The authors have no personal or professional affiliations or relationships that could influence or bias their work.

References

1. Li Y, Tarlow D, Brockschmidt M, Zemel R (2015) Gated graph sequence neural networks. arXiv. https://doi.org/10.48550/ARXIV.1511.05493
2. Lopera DS, Servadei L, Kiprit GN, Hazra S, Wille R, Ecker W (2021) A survey of graph neural networks for electronic design

automation. In: 2021 ACM/IEEE 3rd workshop on machine learning for CAD (MLCAD), pp 1–6. https://doi.org/10.1109/MLCAD52597.2021.9531070
3. Ma Y, He Z, Li W, Zhang L, Yu B (2020) Understanding graphs in EDA: from shallow to deep learning. Association for Computing Machinery, New York, pp 119–126. https://doi.org/10.1145/3372780.3378173
4. Wang H, Yang J, Lee HS, Han S (2018) Learning to design circuits. arXiv. https://doi.org/10.48550/ARXIV.1812.02734
5. Wang H, Wang K, Yang J, Shen L, Sun N, Lee HS, Han S (2020) GCN-RL circuit designer: transferable transistor sizing with graph neural networks and reinforcement learning. arXiv. https://doi.org/10.48550/ARXIV.2005.00406
6. He H, Zhang G (2018) End-to-end learning for distributed circuit design
7. Zhang G, He H, Katabi D (2019) Circuit-GNN: graph neural networks for distributed circuit design. In: Proceedings of the 36th international conference on machine learning. Proceedings of Machine Learning Research, vol 97, pp 7364–7373
8. Hakhamaneshi K, Nassar M, Phielipp M, Abbeel P, Stojanović V (2022) Pretraining graph neural networks for few-shot analog circuit modeling and design. arXiv. https://doi.org/10.48550/ARXIV.2203.15913
9. Gao X, Deng C, Liu M, Zhang Z, Pan D.Z, Lin Y (2021) Layout symmetry annotation for analog circuits with graph neural networks. ASPDAC '21. Association for Computing Machinery, New York, pp 152–157. https://doi.org/10.1145/3394885.3431545
10. Chen H, Zhu K, Liu M, Tang X, Sun N, Pan D.Z (2021) Universal symmetry constraint extraction for analog and mixed-signal circuits with graph neural networks. In: 2021 58th ACM/IEEE design automation conference (DAC), pp 1243–1248. https://doi.org/10.1109/DAC18074.2021.9586211
11. Wang H, Wang K, Yang J, Shen L, Sun N, Lee H.-S, Han S (2020) Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning. In: Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference. DAC '20
12. Wang H, Yang J, Lee H, Han S (2018) Learning to design circuits. CoRR abs/1812.02734 1812.02734
13. Khamis A, Agamy M (2023) Comprehensive mapping of continuous/switching circuits in CCM and DCM to machine learning domain using homogeneous graph neural networks. IEEE Open J Circuits Syst. https://doi.org/10.1109/OJCAS.2023.3234244
14. Xie J, Alvarez-Fernandez I, Sun W (2020) A review of machine learning applications in power system resilience. In: 2020 IEEE power and energy society general meeting (PESGM), pp 1–5. https://doi.org/10.1109/PESGM41954.2020.9282137
15. Ye Y, Li Y, Ouyang R, Zhang Z, Tang Y, Bai S (2023) Improving machine learning based phase and hardness prediction of high-entropy alloys by using gaussian noise augmented data. Comput Mater Sci 223:112140. https://doi.org/10.1016/j.commatsci.2023.112140
16. Goodrick K.J, Butler A, Byrd,T. Maksimović D (2021) Machine learning estimators for power electronics design and optimization, pp 1–8 https://doi.org/10.1109/DMC51747.2021.9529937
17. Li X, Zhang X, Lin F, Blaabjerg F (2022) Artificial-intelligence-based design for circuit parameters of power converters. IEEE Trans Ind Electron 69(11):11144–11155. https://doi.org/10.1109/TIE.2021.3088377
18. Khamis A.K, Agamy M (2022) Mapping continuous circuit structures to machine learning space. In: 2022 IEEE 31st international symposium on industrial electronics (ISIE), pp 149–155. https://doi.org/10.1109/ISIE51582.2022.9831581
19. Khamis A, Agamy M (2022) Converter circuits to machine learning: optimal feature selection. In: 2022 IEEE energy conversion congress and exposition (ECCE), pp 1–7. https://doi.org/10.1109/ECCE50734.2022.9947826
20. Khamis AK, Agamy M (2024) Circuit dynamics prediction via graph neural network and graph framework integration: three phase inverter case study. IEEE Open J Power Electron. https://doi.org/10.1109/OJPEL.2024.3416195
21. Mo L, Chen G, Huang J, Qing X, Hu Y, He X (2022) Graph theory-based programmable topology derivation of multiport DC-DC converters with reduced switches. IEEE Trans Ind Electron 69(6):5745–5755. https://doi.org/10.1109/TIE.2021.3090711
22. Delgado M, Profos G (1999) Identifiability of dynamic systems represented by bond graphs. Math Comput Modell Dyn Syst 5(2):89–112. https://doi.org/10.1076/mcmd.5.2.89.6174
23. Ma Y, He Z, Li W, Zhang L, Yu B (2020) Understanding graphs in EDA: from shallow to deep learning. Association for Computing Machinery, New York, pp 119–126. https://doi.org/10.1145/3372780.3378173
24. Ustun E, Deng C, Pal D, Li Z, Zhang Z (2020) Accurate operation delay prediction for FPGA HLS using graph neural networks. In: 2020 IEEE/ACM international conference on computer aided design (ICCAD), pp 1–9
25. Ren H, Kokai GF, Turner WJ, Ku TS (2020) Paragraph: layout parasitics and device parameter prediction using graph neural networks. In: Proceedings of the 57th ACM/EDAC/IEEE design automation conference. IEEE Press
26. Li Y, Lin Y, Madhusudan M, Sharma A, Xu W, Sapatnekar SS, Harjani R, Hu J (2020) A customized graph neural network model for guiding analog IC placement. In: 2020 IEEE/ACM international conference on computer aided design (ICCAD), pp 1–9
27. Umarikar A, Umanand L (2005) Modelling of switched mode power converters using bond graph. Electr Power Appl IEE Proc 152:51–60. https://doi.org/10.1049/ip-epa:20040828
28. Gebben VD (1977) Bond graph bibliography for 1961–1976. J Dyn Syst Measur Control 99(2):143–145
29. Markakis,A, Holderbaum W, Potter B (2011) A comparison between bond graphs switching modelling techniques implemented on a boost dc-dc converter. In: 2011 IEEE 33rd international telecommunications energy conference (INTELEC). https://doi.org/10.1109/INTLEC.2011.6099778
30. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
31. Velickovic P, Cucurull G, Casanova A, Romero A, Li'o P, Bengio Y (2018) Graph attention networks. arXiv preprint arXiv:1710.10903
32. Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. arXiv. https://doi.org/10.48550/ARXIV.1706.02216
33. Schlichtkrull M, Kipf TN, Bloem P, Berg RVD, Titov I, Welling M (2018) Modeling relational data with graph convolutional networks. arXiv preprint arXiv:1703.06103