IEEE Open Journal of
Industry Applications

Received 7 February 2023; revised 13 May 2023 and 8 June 2023; accepted 13 June 2023. Date of publication 3 July 2023;
date of current version 13 July 2023. The review of this article was arranged by Associate Editor Luca Solero.

Digital Object Identifier 10.1109/OJIA.2023.3291637

Sequence Prediction for SiC MOSFET Active Gate Driving With a Recurrent Neural Network

LI YANG [1], YUXUAN LIU[2], WENSONG YU [1] (Member, IEEE), AND IQBAL HUSAIN [1] (Fellow, IEEE)

[1]FREEDM System Center, North Carolina State University, Raleigh, NC 27606 USA
[2]Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC 27606 USA

CORRESPONDING AUTHOR: Li Yang (lyang20@alumni.ncsu.edu)

This work was supported by Texas Instruments Inc.

ABSTRACT This article develops a recurrent neural network (RNN) with an encoder–decoder structure to predict the driving sequence of SiC MOSFET active gate drivers (AGDs). With a set of switching targets as the input, the predictor generates an optimal active gate driving sequence to improve the switching transient. The development is based on a hybrid platform across MATLAB, PyTorch, and LTspice. A high-fidelity switching model is implemented in MATLAB to obtain reliable training data. The sequence predictor is trained with PyTorch. The predicted sequence is verified on an example Buck circuit in LTspice. In contrast to the state-of-the-art approach, the proposed method avoids exhaustive search in a large solution space; the sequence length is dynamically predicted per the driving strength at each step. The AGD sequences generated by the predictor effectively and precisely improve the switching transients, making the proposed sequence predictor an integral and valuable component for active gate driving.

INDEX TERMS Active gate driver (AGD), deep learning, recurrent neural network (RNN), sequence prediction, SiC MOSFET.

I. INTRODUCTION

Despite the benefits of SiC MOSFETs in building high efficiency, high power density, and high-performance power conversion systems, the high switching speed also causes more significant overshoot, oscillation, and elevated electromagnetic interference (EMI). Active gate driver (AGD) is a remedy for adopting SiC devices while addressing problems they introduce.

Although extensive efforts have been made on AGD circuit implementations [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], a methodology for designing the active driving sequence still needs to be developed. To address the challenge, a search approach is proposed in [9] and is improved in [13] where a dynamic resistance sequence with 60 time steps is searched by the particle swarm optimization (PSO) algorithm. Another model-based solution was proposed in [14] to find the optimal resistance sequence for the AGD reported in [8]. The optimal gate resistance value adopted for oscillation reduction is found by trial and error.

As shown in Fig. 1, the state-of-the-art method initializes a group of candidate AGD sequences with predefined fixed lengths [9], [13]. Then, the AGD sequences are applied to the physical circuit, and switching results of interest are captured. The actual circuit is regarded as an AGD to switching results (SR) model, which a high-fidelity simulation platform can also implement. The PSO algorithm takes the SR model's feedback to update the candidate AGD sequences and iteratively searches for the optimal solution. The fixed sequence length assumption of the state-of-the-art method is of concern.

In this article, an AGD sequence prediction method is developed to address the challenge, as shown in Fig. 1. The total length and individual driving strengths are predicted, and their values are dynamically matched. Instead of searching from candidate AGD sequences, which can be poorly defined, the predictor takes switching targets as input and generates the driving sequences. The predictor uses the recurrent neural network (RNN) to process the sequential data effectively. The proposed method can fine-tune the switching transient parameters to reduce the switching loss or improve the EMI profile.

The rest of the article is organized as follows. The principle of RNN-based sequence prediction is given in Section II, training data generated from a high-fidelity switching model

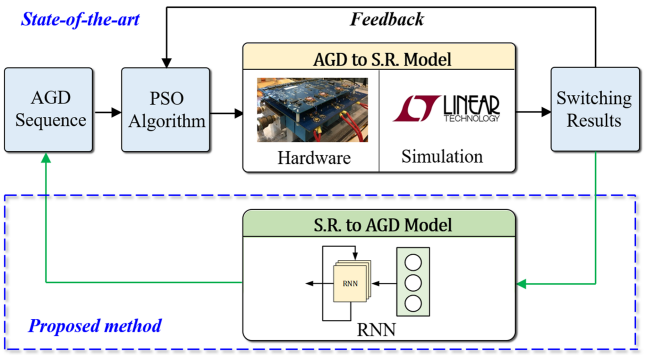

FIGURE 1. Working principle of AGD sequence searching method based on PSO algorithm [13] in conjunction with the proposed prediction method based on RNN.

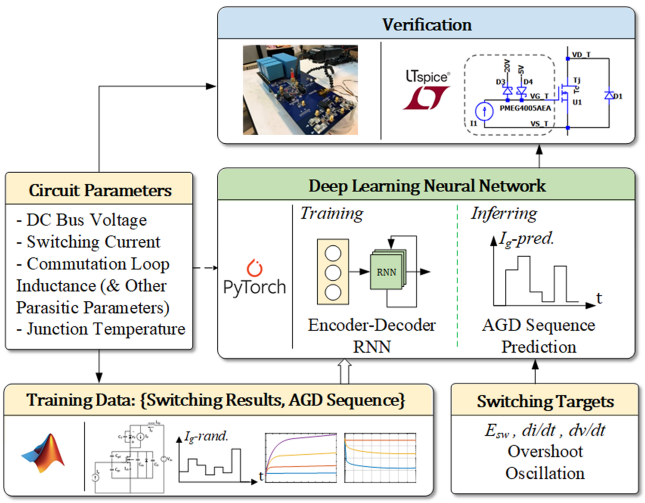

FIGURE 2. Data-driven workflow using an encoder–decoder recurrent neural network (ED-RNN).

is discussed in Section III, the neural network structure and the training process are elaborated in Section IV, and the AGD performance verification is given in Section V. Finally, Section VI concludes the article.

II. RECURRENT NEURAL NETWORK-BASED SEQUENCE PREDICTION

The hybrid data-driven workflow for AGD sequence prediction is demonstrated in Fig. 2. The deep-learning neural network developed for the task is at the workflow's core. A predictor that models the long-term dependency within a time series is critical for the prediction problem. For example, to generate a driving sequence leading to a particular switching loss, the predictor must oversee the loss generated in the previous steps and then, predict the present step. Since tradeoffs always exist among multiple switching targets in one switching transient, understanding long-term dependencies by the predictor is essential. The encoder–decoder RNN (ED-RNN) presented in this article is developed to solve the

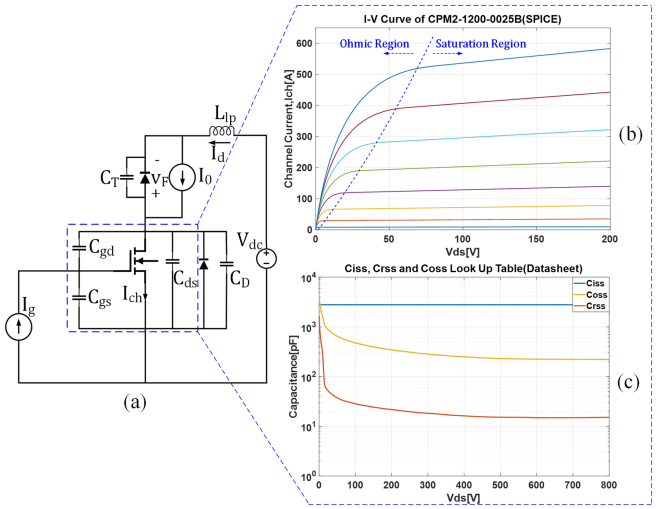

FIGURE 3. (a) The DPT circuit modeled in MATLAB for data generation. (b) The I-V characteristic curve extracted from the device SPICE model. (c) The C-V characteristic curve extracted from the device datasheet.

time-series modeling problem; the structure of the ED-RNN will be elaborated in Section IV.

The training data for the ED-RNN consists of AGD sequences paired with their corresponding switching results, including $E_{sw}, di/dt, dv/dt$, overshoot, and oscillations. In this work, a high-fidelity switching model is developed in MATLAB where large numbers of switching transients can be quickly simulated and quantified. The switching results are treated as the inputs to the ED-RNN, and the AGD sequence prediction is the expected output.

The implementation and training of the ED-RNN are based on PyTorch, a high-performance integrated library for AI applications [27]. Once the neural network is trained, it makes the AGD sequence prediction, referred to as the inferring process. The ED-RNN predictor takes user-defined switching targets as the input and predicts a driving sequence, which is verified in the LTspice platform.

The hybrid workflow enables a cross-verification of the proposed method. The training data generation and AGD performance verification are conducted on different platforms, hence, the AGD predictor is a generalized approach and not platform dependent.

III. HIGH FIDELITY SWITCHING MODEL FOR NN TRAINING

A. DATA GENERATION MODEL

The training data can be obtained from a circuit as simple as a double pulse tester (DPT). A DPT circuit is modeled in MATLAB script for training data generation, shown in Fig. 3(a). Die models of a SiC MOSFET and a diode are used to eliminate the influences of packaging parasitic inductance. The parasitic driving loop inductance is neglected, while the commutation loop inductance (L_{LP}) is retained. The dc bus voltage (V_{DC}) and switching current (I_0) are fixed for the

TABLE 1. Parameters of the DPT Switching Model for Data Generation

Parameters	Value
MOSFET	CPM2-1200-0025B(CREE)
Diode	CPW41200S020B(CREE)
V_{DC}	800 V
I_0	40 A
L_{LP}	20 nH
Current source AGD	$I_g \in I_{dv} = \{i \mid i = 0.1 : 0.1 : 2.0\}$ @ 3 ns $-5\ V \leq V_{dv} \leq 20\ V$

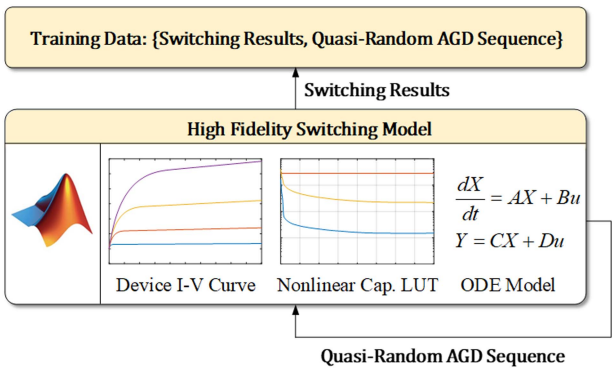

FIGURE 4. High fidelity data generation workflow.

model but can be modified to obtain switching data on different operating conditions. An active gate current (I_g) driver is considered in this work and is assumed to generate discrete currents from $i = 0.1$ A to $i = 2.0$ A with 20 levels. The time step for I_g is selected based on the switching speed of the target SiC MOSFET. On the one hand, a shorter time step is preferred to fine-tune the switching transient of the device, but on the other, it should be long enough for the active I_g to alter I_{ds} and V_{ds} at each step. Given that the turn-ON time of the device is typically 30–80 ns [15], 3 ns time step is a proper choice to have around ten steps in the driving sequence even for the fastest switching transient. The circuit parameters are listed in Table 1.

B. HIGH FIDELITY DATA GENERATION

Fig. 4 highlights the high-fidelity data generation workflow. Three methods are applied to obtain reliable data from the switching model: the device I-V curve, the capacitance C-V look-up table (LUT), and the ordinary differential equation (ODE) models. Quasi-random AGD sequences are generated, serving as the switching model excitations.

1) DEVICE I-V CURVE

The characteristic $I_{ch} - V_{ds}$ curve of the CPM2-1200-0025B is critical to describe the output behavior of the SiC MOSFET. The curve is usually available on the device datasheet, but the V_{ds} range is limited and insufficient to model the device entirely. This work uses the SPICE model developed by the device manufacturer to extract the extended $I - V$ curve. The $I - V$ curve of CPM2-1200-0025B obtained from the manufacturer SPICE model at 25 °C junction temperature is shown in Fig. 3(b).

In the ohmic region, the channel current (1), as proposed in [16], is adopted to approximate the $I - V$ curve.

$$I_{ch} = \frac{\beta_{ohm}}{1 + \frac{\beta_{ohm} V_{ds}}{g_{sat}}} \left(V_{gs} - V_{TH} - \frac{k}{2} V_{ds} \right) V_{ds} \tag{1}$$

where I_{ch} is the channel current of the MOSFET, V_{ds}, V_{gs}, V_{TH} are the drain-source, gate-source, and threshold voltages, respectively. The β_{ohm}, g_{sat}, and k are device-related parameters and are obtainable by fitting the device $I - V$ curves using (1). Note that (1) differs from the current equation for Si MOSFETs.

The channel current equation for the Saturation region is

$$I_{ch} = \frac{\beta_{sat}}{2} (V_{gs} - V_{TH})^2 (1 + \lambda V_{ds}) \tag{2}$$

where λ is channel length modulation coefficient, β_{sat} is a device parameter. These device parameters are obtained by fitting the $I - V$ curve in the saturation region.

The MOSFET $I - V$ characteristic is given as a group of $I_{ch} - V_{ds}$ curves under different V_{gs} gate voltages. For a particular $I_{ch} - V_{ds}$ curve, the device parameters (β_{ohm}, g_{sat}, k, β_{sat}, λ) can be obtained by curve fitting using (1) and (2). Therefore, for multiple $I - V$ curves under different V_{gs}, the device parameters are formulated as functions to V_{gs}. A boundary voltage V_{ct} is also formulated such that the MOSFET is in the ohmic region if $V_{ds} < V_{ct}$; it is in the saturation region when $V_{ds} > V_{ct}$. The curve of V_{ct} is shown in Fig. 3(b).

2) CAPACITANCE C-V LUT

The conventional method for modeling the nonlinear capacitances of MOSFETs, i.e., C_{iss}, C_{rss}, C_{oss} as shown in Fig. 3(c), is done by curve-fitting the capacitances to theoretical equations [17], [18]. However, the curve-fitting results cannot precisely match the experimental values of the capacitances, especially in the low V_{ds} range where the capacitances vary remarkably. The mismatch in the nonlinear capacitances reduces the fidelity of the data generation. As discussed in the paper, the DPT circuit dynamics is described by ODEs and solved numerically. Under this framework, LUT-based implementation of the capacitance $C - V$ curves can achieve better accuracy. By adopting a small time step to solve V_{ds} and updating the capacitance values accordingly, the switching model incrementally incorporates the nonlinear capacitances into the transient simulation model without using any explicit equation.

3) ODE MODEL

The DPT circuit is modeled by its differential equations to obtain accurate switching transient data and solved numerically by MATLAB. Unlike behavioral modeling methods such as

in [17], [19], [20] where certain assumptions are made to derive the analytical equations, the numerical solution based on the circuit ODE model requires the fewest assumptions while retaining the nonlinearities, which provides better accuracy.

The turn-ON process is divided into four modes and named in the format "x - y," where "x" is the state of the MOSFET, and "y" is the state of the diode. Starting from the OFF–ON mode where the MOSFET is OFF and the diode is free-wheeling. The gate current I_g is charging the C_{iss} and brings V_{gs} up from $V_{EE} = -5$ V to V_{TH}. In this mode, no dynamics in the commutation loop are involved. The gate voltage equation is given below where $C_{iss} = C_{gs} + C_{gd}$

$$V_{gs} = V_{EE} + \frac{I_g}{C_{iss}} t. \tag{3}$$

At the moment the MOSFET turns ON, $V_{ds} = V_{DC}$, and hence, the MOSFET is in the saturation region, and the turn-ON process is in the Sat-On mode. The Sat-On mode ends when I_d reaches the load current I_0 as in Fig. 3(a), and the diode turns OFF (Sat-Off). The gate driver loop equation is shown in (4)

$$\frac{I_g}{C_{gs}} t + \frac{C_{rss}}{C_{gs}} V_{ds} - \frac{C_{iss}}{C_{gs}} V_{od} - \left[\frac{C_{rss}}{C_{gs}} V_{ds}^0 - \frac{C_{iss}}{C_{gs}} V_{od}^0 \right] = 0 \tag{4}$$

where $V_{od} = V_{gs} - V_{TH}$ is the overdrive voltage. V_{ds}^0 and V_{od}^0 are initial values of V_{ds} and V_{od} in each simulation time step. The commutation loop voltage and current equations are shown in (5) and (6), the parameters are shown in Fig. 3(a) and $C_{rss} = C_{gd}$, $C_B = C_{oss} + C_D$. Here, V_f is the forward voltage drop of the SiC diode CPW41200S020B. Its value to the forward current is curve-fitted by the datasheet information as $V_f = 0.03194 I_f + 0.9205$.

$$V_{DC} + V_f - V_{ds} - L_{LP} \frac{dI_d}{dt} = 0 \tag{5}$$

$$\frac{\beta_{sat}}{2} V_{od}^2 (1 + \lambda V_{ds}) = I_d + C_{rss} \frac{dV_{od}}{dt} - C_B \frac{dV_{ds}}{dt}. \tag{6}$$

Generally, the actual switching loss E_{sw} caused by the MOSFET channel current I_{ch} is immeasurable; the measured switching loss E_{me} is calculated by the drain current I_d, but is an underestimation of the actual loss [21]. One feature of the proposed modeling approach is that the actual switching loss E_{sw} and the measured switching loss E_{me} can be differentiated as described in (7) and (8). Both E_{sw} and E_{me} can be obtained, but only E_{sw} is used as the training data in this work.

$$\frac{dE_{sw}}{dt} = I_{ch} V_{ds} = \frac{\beta_{sat}}{2} V_{od}^2 (1 + \lambda V_{ds}) V_{ds} \tag{7}$$

$$\frac{dE_{me}}{dt} = I_d V_{ds}. \tag{8}$$

The Sat-Off mode starts when the diode turns OFF and ends when the MOSFET transitions from the saturation region to the ohmic region (Ohm-Off). The commutation loop voltage equation for this mode changes from (5) to (9), where C_T is the parasitic capacitance of the top switch and $C_T = C_{oss} + C_D$. Meanwhile, (4), (6), and (7) remain unchanged.

$$\frac{dV_{ds}}{dt} + \frac{1}{C_T} I_d + L_{LP} \frac{d^2 I_d}{dt^2} - \frac{I_0}{C_T} = 0. \tag{9}$$

In the Ohm-Off mode, the circuit begins oscillating. As V_{gs} increases till V_{CC}, the turn-ON process finishes, and the MOSFET reaches its steady operating point in the ohmic region. Commutation loop current (6) and loss (7) are changed to (10) and (11), while (4) and (9) remain the same.

$$\frac{\beta_{ohm}}{1 + \frac{\beta_{ohm}}{g_{sat}} V_{ds}} \left(V_{od} - \frac{k}{2} V_{ds} \right) V_{ds} = I_d + C_{rss} \frac{dV_{od}}{dt} - C_B \frac{dV_{ds}}{dt} \tag{10}$$

$$\frac{dE_{sw}}{dt} = \frac{\beta_{ohm}}{1 + \frac{\beta_{ohm}}{g_{sat}} V_{ds}} \left(V_{od} - \frac{k}{2} V_{ds} \right) V_{ds}^2 \tag{11}$$

The switching transient is modeled according to the gate driver loop equation, the commutation loop voltage and current equations, and the switching loss equation. The flowchart for the switching model is summarized in Fig. 5. The simulation time step 0.1 ns and the AGD time step 3 ns are example values used for this work and can be changed.

C. QUASI-RANDOM AGD SEQUENCE

On the data generation platform, I_g sequences are randomly generated and concurrently applied to the turn-on transient. In this work, the I_g sequences are generated by *randsample*() function that accepts a probability distribution input. For the turn-on transient, the quasi-random I_g sequences during the current rising interval are generated with probability distribution $P(di/dt)$, and that for the voltage falling interval is $P(dv/dt)$. The proposed switching model makes the turn-ON transient a white box, and hence, it is convenient to assign unique I_g sequences for the current rising and voltage falling intervals individually.

It should be noted that the quasi-random AGD sequences are generated in parallel with the switching transient simulation rather than being predefined. When the switching transient finishes, the AGD sequence generation also terminates. Therefore, the sequence length is dynamically matched with the duration of the transient.

D. TRAINING SET RESULTS

The switching model is implemented with MATLAB. m script, where the optimized ODE solver, *ode45()*, generates the numerical solutions. An efficient search algorithm is developed for the nonlinear capacitance $C - V$ LUT to run the switching model. The circuit ODEs presented before are transformed to state space form for the *ode45()* solver. The simulation runs and updates the coefficients of the ODEs (such as the nonlinear capacitance values) at every 0.1 ns. The small time step guarantees the high accuracy of the switching model. The proposed switching model generates approximately 15 000 training data, which takes about 4–5 h.

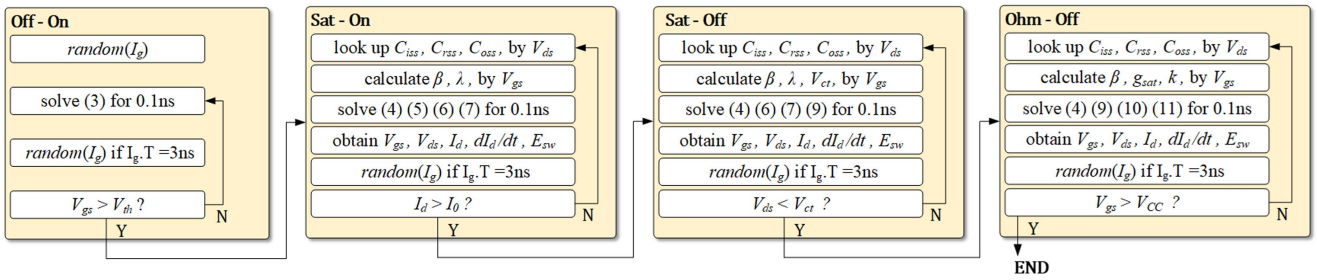

FIGURE 5. Data generation algorithm flowchart.

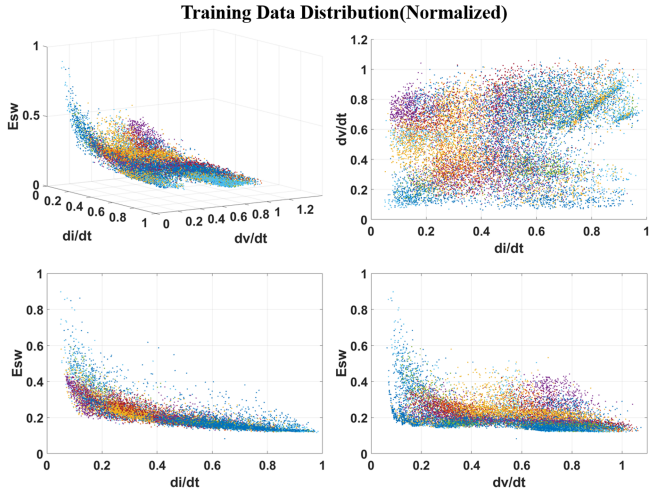

FIGURE 6. Obtained training data from the switching model (3D and 2D views).

Three switching results are presented here as an example: $E_{sw}-$ the switching loss, $di/dt-$ the average current slope on the rising edge, and $dv/dt-$ the average voltage slope on the falling edge. The normalized switching results distribution with di/dt, dv/dt and E_{sw} as x, y, and z axes is visualized in Fig. 6. It is clear that on the $x-y$ plane, the switching results appear over the entire plane. In other words, most of the possible di/dt, dv/dt, and E_{sw} combinations have been included in the training data, making the training more reliable. Here, the key enabler is the quasi-random I_g generation method with variable di/dt and dv/dt. It should also be mentioned that AGD sequences generate the switching results denoted by the same color with the same probability distribution. The base values used for the normalization are $E-$ the switching loss under constant $I_g = 0.1$ A; K_i, K_v- the average current and voltage slopes when constant $I_g = 2.0$ A is applied.

In summary, the SiC MOSFET model is developed based on the device $I-V$ characteristic, and the capacitance $C-V$ characteristic is implemented by LUT from measured data instead of fitting equations. The switching model is based on ODEs, which are more accurate than behavioral models based on analytical equations. The variables di/dt and dv/dt are state variables in the switching model and can be easily solved. The AGD sequences can be generated individually for di/dt and dv/dt stages using the white-box switching model. The probability distributions assigned to the sequences can be manually optimized, which is a unique and helpful feature for high-quality training data generation.

IV. GRU-BASED ENCODER-DECODER RECURRENT NEURAL NETWORK

The gated recurrent unit (GRU) network is one of the two main variants of classical RNN that can process time series with long time dependencies [22]. For this reason, GRU is adopted here to construct the active gate driving sequence predictor. The sequence predictor also takes advantage of the encoder–decoder network structure successfully applied to machine translation [23] and image autocaptioning [24] problems. Fig. 7 demonstrates the overall structure of the GRU-based encoder–Decoder recurrent neural network (GRU-EDRNN).

A. ENCODER AND DECODER STRUCTURE

The encoder takes the switching results as input and generates a context vector C for the decoder. The encoder consists of two components: 1) batch normalization; and 2) linear layer.

The batch normalization (BN) algorithm [25], [26] is to whiten the training data within a minibatch input. It helps the neural network learn the optimal distribution of the training data, making the training faster and smoother. This work uses a BN layer as the front end to normalize input data properly.

The linear layer is defined as follows:

$$C(x) = \mathcal{A}(Wx + b) \tag{12}$$

where x is the vector of switching results, C is the context vector, and W and b are the learnable weight matrix and bias vector. The $\mathcal{A}$ is a linear activation function in this work.

The decoder takes the context vector and decodes it as the AGD sequence. The decoder consists of five parts as follows.

1) embedding layer;
2) GRU;
3) dropout;
4) BN and linear layer;
5) softmax layer.

As mentioned above, at each time step, the active I_g can be chosen from 0.1 to 2.0 A with 20 levels. Two unique

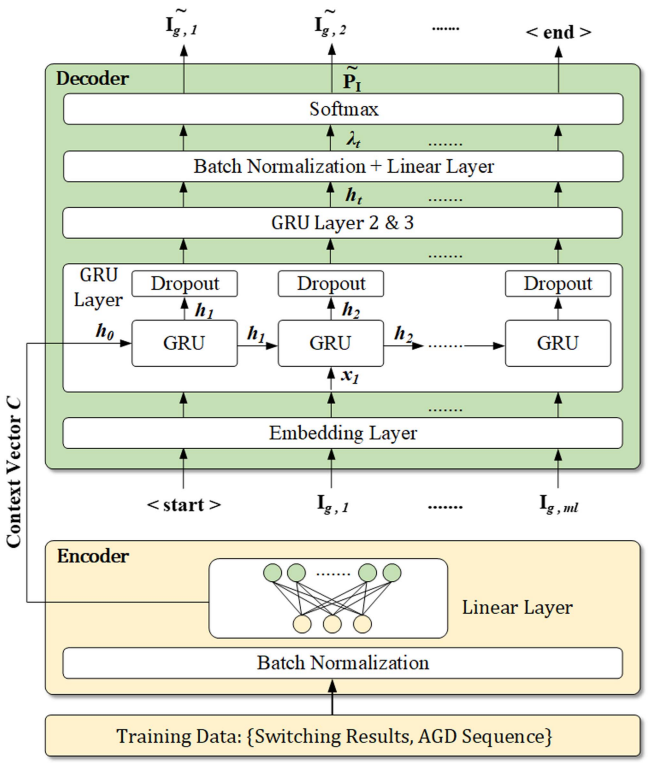

FIGURE 7. GRU9based encoder–decoder neural network.

tokens, $< start >$ and $< end >$, are manually added to the first and last positions to mark the starting and completing of an AGD sequence. There are 22 selections for each time step of an AGD sequence. In this work, an embedding layer [28] is adopted to encode the AGD sequence.

The context vector C is then used as the input h_0 to the first GRU cell in the decoder. GRU is capable of "forgetting" irrelevant information and "memorizing" long-time dependencies in a sequence due to the reset and update gate mechanism. Three GRU layers are cascaded as shown in Fig. 7; in this way, the long-time dependency modeling capability is enhanced.

Dropout is an effective method to overcome the overfitting problem in a neural network [29], hence, it helps the neural network generalize better.

The AGD sequence prediction is a classification problem, in essence, where for each time step, there are 22 classes. The neural network treats its output as a probability distribution, which indicates the probability of I_g equaling any of the 22 classes as follows:

$$\widetilde{P_I} = [P_{\langle start \rangle}, P_{0.1}, P_{0.2}, \ldots, P_{2.0}, P_{\langle end \rangle}] \tag{13}$$

With the probability distribution, the class with the highest probability is chosen as the prediction result. The Softmax layer converts the linear layer output λ_t to the probability distribution $\widetilde{P_I}$.

B. NEURAL NETWORK TRAINING

Consider a specific data set containing AGD sequence l of length m_l: $\{I_{g,k} | k = 1, \ldots, m_l\}$ and the corresponding switching results vector x. Assuming $I_{g,0} =< start >$ and $I_{g,m_l+1} =< end >$, the goal of training is for the neural network to learn the prediction: $I_{g,0,\ldots,k-1} \rightarrow I_{g,k}$, where $I_{g,0,\ldots,k-1}$ means sequentially inputting $I_{g,0}, I_{g,1}, \ldots, I_{g,k-1}$ to the GRU-EDRNN.

In practice, at step k where $C(x)$, $I_{g,0,\ldots,k-1}$ are the inputs to the neural network, the GRU-EDRNN makes a prediction for $I_{g,k}$ denoted as $\widetilde{I_{g,k}}$. At each step, the neural network output is a probability distribution $\widetilde{P_I}$ as in (13), the $\widetilde{I_{g,k}}$ is obtained by

$$\widetilde{I_{g,k}} = np.\text{argmax}(\widetilde{P_I}) \tag{14}$$

where $np.argmax()$ is a Python function to return the index of the maximum element.

In the same way, $I_{g,k}$ can be regarded as a probability distribution $P_I = [0.0, \ldots, 0.0, 1.0, 0.0, \ldots, 0.0]$ where the 1.0 appears at the kth position. Therefore, to make the correct prediction $\widetilde{I_{g,k}} = I_{g,k}$, $\widetilde{P_I}$ should be as close to P_I as possible. Mathematically, the distance between two probability distributions is measured by cross-entropy and is formulated as follows:

$$\begin{aligned} \mathcal{H}(P_I, \widetilde{P_I}) &= -\sum_{i=1}^{22} P_I(i) \log \widetilde{P_I}(i) \\ &= -\log[P(\widetilde{I_{g,k}} = I_{g,k})]. \end{aligned} \tag{15}$$

Therefore, the following optimization problem over the entire training set explains the training process. The W, U, and b are the learnable parameters of the neural network, N is the total number of training sets, and m_l is the length of AGD sequence l.

$$\begin{aligned} (\tilde{W}, \tilde{U}, \tilde{b}) = \text{argmin}_{W,U,b} - \frac{1}{N} \sum_{l=1}^{N} \frac{1}{m_l + 1} \sum_{k=1}^{m_l+1} \\ \log[P(\widetilde{I_{g,k}^{(l)}} = I_{g,k}^{(l)}]. \end{aligned} \tag{16}$$

To summarize, for a sequence $l : \{< start >, I_{g,1}, I_{g,2}, \ldots, I_{g,m_l}\}$ along with its switching results x, the expected prediction is $q : \{I_{g,1}, I_{g,2}, \ldots, I_{g,m_l}, < end >\}$, the actual prediction is $\widetilde{q} : \{\widetilde{I_{g,1}}, \widetilde{I_{g,2}}, \ldots, \widetilde{I_{g,m_l}}, \widetilde{I_{g,m_l+1}}\}$. The training algorithm calculates the cross-entropy losses between the elements in q and $\widetilde{q}$. After training, the weights W, U, and b are optimized, and the sequence predictor model is obtained.

C. NEURAL NETWORK INFERRING

Inferring refers to the operation when the neural network has been trained and is then used to make the AGD sequence prediction. The inferring process is demonstrated in Fig. 8. The input to the neural network is the switching targets of interest. In practice, the <*start*> token is one additional input

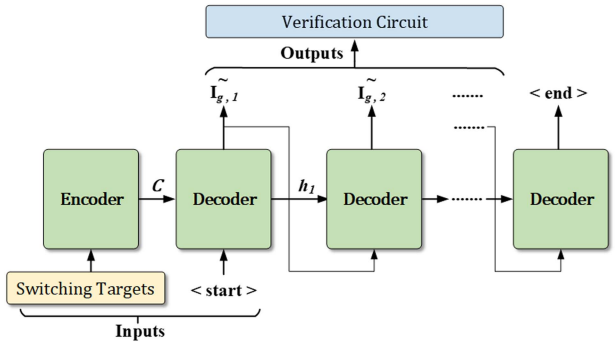

FIGURE 8. Inferring process.

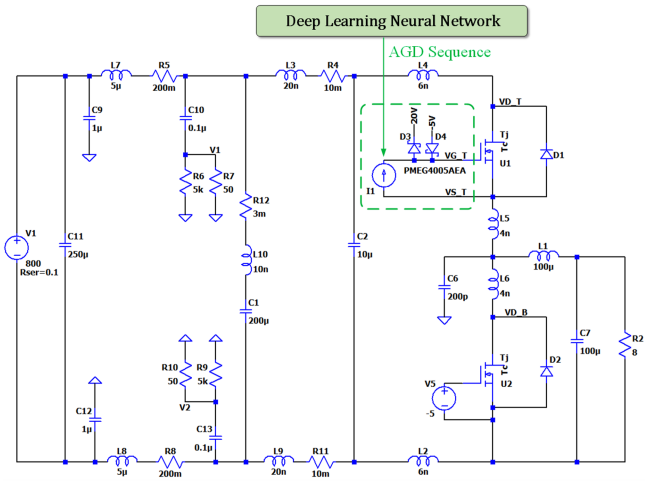

FIGURE 9. Buck circuit implemented in LTspice with current source AGD, where the I_g sequence is predicted by the GRU-EDRNN.

to the neural network to start the prediction. Once the first step prediction is made, it will be used as the input to the decoder and generates the second step prediction. The process is repeated until the neural network predicts the <*end*> token, indicating that a complete AGD sequence has been found. The neural network infers the sequence length and is dynamically changed for different switching targets.

It should be mentioned that the duration of the training and inferring processes relies on the hardware used. This work utilizes Paperspace (https://www.paperspace.com/), a cloud infrastructure for machine learning applications. Due to the neural network's complexity, the training data's size, and the GPU infrastructure in use, a single training process typically takes around 30 min (with multiple runs required to obtain the final model). Conversely, inferring usually takes only a couple of seconds.

V. AGD PREDICTOR PERFORMANCE VERIFICATION

The verification of the AGD predictor is based on a Buck converter with an active current source gate driver. The GRU-EDRNN predicts AGD sequences to improve the switching transients. Fig. 9 shows the verification circuit in LTspice with

TABLE 2. Switching Transient Improvement Results by AGD

Case	Item	E_{sw}	di/dt	dv/dt
	Base value	$4000\mu J$	$8.3kA/\mu s$	$-82.7kV/\mu s$
CGD 1	Achieved	0.3000	0.2572	0.4564
AGD 1	Target	0.2700	0.3000	0.3900
	Achieved	0.2738	0.2856	0.3915
AGD 2	Target	0.3700	0.2600	0.3600
	Achieved	0.3773	0.2606	0.3629
AGD 3	Target	0.2380	0.3530	0.4520
	Achieved	0.2412	0.3534	0.4507

the ideal current source as the AGD. The dc bus is 800 V, and the turn-ON current is 40 A. Total commutation loop inductance is 20 nH. The circuit parameters for the verification is the same as the switching model where the training data are generated. An example total base plate to ground coupling capacitance of 200 pF is adopted [30], which is the source of common mode noise. Two line impedance stabilization networks (LISN) are added to DC+ and DC–. By computing $V_{CM} = \frac{1}{2}(V_1 + V_2)$ as labeled on the LISNs, the common mode noise voltage is evaluated. The CGD with a constant gate resistor is used as the benchmark. The circuit configurations are the same as in Fig. 9, except that voltage source CGD with a constant gate resistor R_g replaces the current AGD in the dashed box.

A. SWITCHING WAVEFORM MODIFICATION BY AGD

This section discusses how AGD modifies the switching transient, focusing on the tradeoff between switching loss E_{sw} and common mode noise V_{CM}. The switching targets for the optimization contain $E_{sw}-$ the switching loss during turn-ON transient; $di/dt-$ the average current rising slope; $dv/dt-$ the maximum voltage falling slope. The AGD will adjust the maximum voltage slope to prevent high common mode noise. Three cases are verified as listed in Table 2.

1) CASE 1

The current source AGD reduces E_{sw} and V_{CM} simultaneously, but di/dt increases as a tradeoff. Table 2 lists the normalized switching results $(E_{sw}, di/dt, dv/dt)$ that the CGD-1 achieves with R_g of 25 Ω. In Case 1, the switching targets for E_{sw} and dv/dt are decreased compared to the CGD-1 case, while that for di/dt increases.

2) CASE 2

For applications emphasizing V_{CM} reduction while having higher tolerance on E_{sw}, the AGD reduces the maximum dv/dt. In this case, the di/dt remains unchanged, and the V_{CM} is remarkably reduced at the cost of enlarged switching loss.

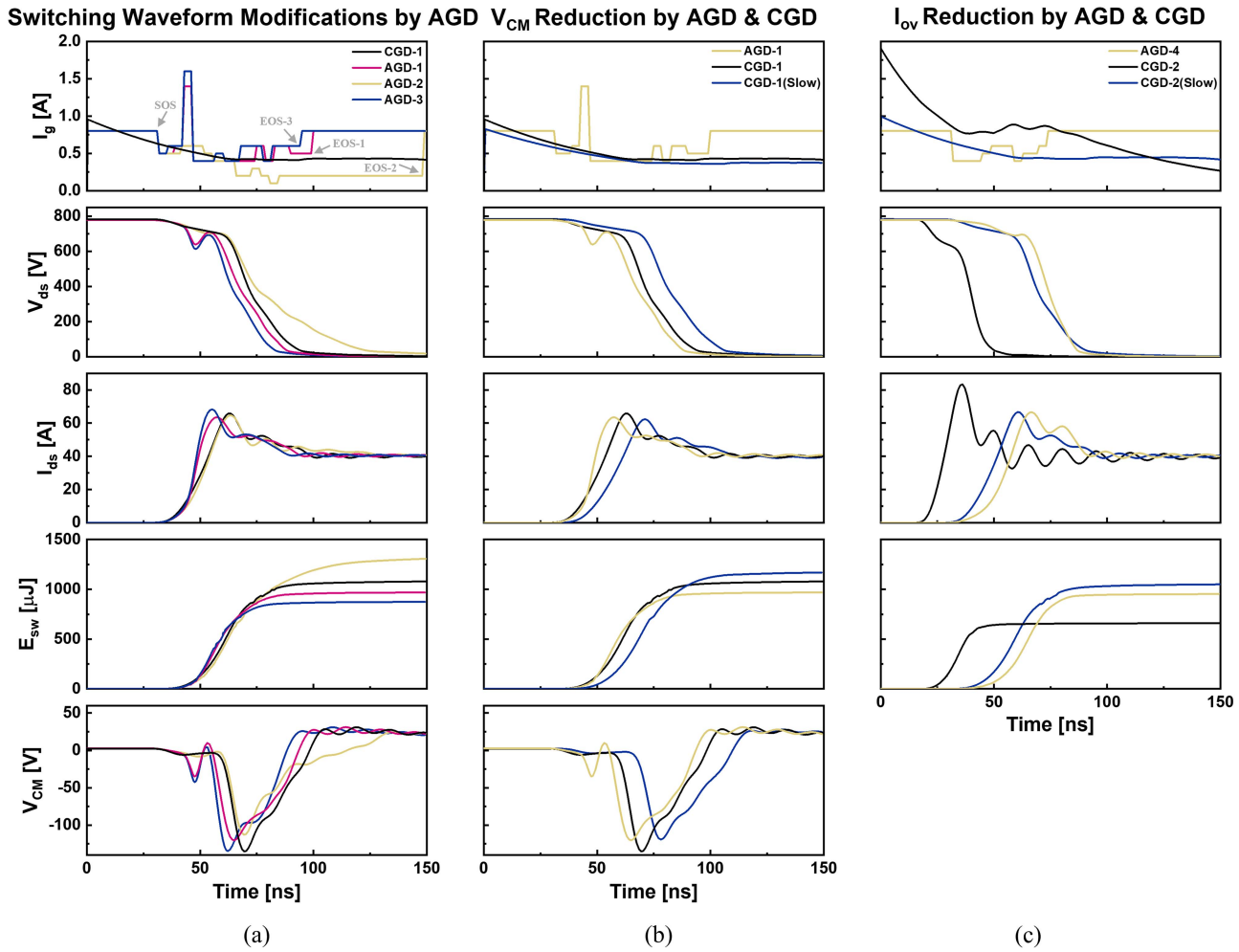

FIGURE 10. (a) Switching transient modification by AGD; (b) V_{CM} reduction by AGD and CGD; (c) I_{OV} reduction by AGD and CGD.

3) CASE 3

For applications with E_{sw} minimization of the dominant target, the AGD optimizes the switching loss without deteriorating the common mode noise. This target is achieved by minimizing the E_{sw}, raising the di/dt, and keeping dv/dt unchanged.

The *Target* shown in Table 2 are the intended switching results to achieve and are normalized values to the *Base Value*. They serve as the inputs to the GRU-EDRNN (Fig. 8). The AGD sequence is applied to the Buck circuit (Fig. 9) once it is predicted. The *Achieved* denotes the E_{sw}, di/dt, dv/dt results by active gate driving. It is observed that the *Achieved* are accurate approximations to the *Target*. The prediction process is a translation from switching targets to AGD sequence, and the results thus confirm the validity of GRU-EDRNN.

The switching waveforms obtained from LTspice are compared in Fig. 10(a) In the I_g comparison waveforms, the CGD-1 gate current starts from 0 ns. For the AGD results, the active gate current sequence all start from 31 ns, indicated by the Start-of-Sequence (SOS) mark. The End-of-Sequence (EOS) marks for the three AGD sequences are also labeled in the figure. Here, the AGD sequences are only generated for the primary switching transient corresponding to the Sat-On and Sat-off modes of the switching model. The turn-on process between 0 ns to SOS corresponds to the OFF–ON mode, where no switching dynamics are involved. A constant $I_g = 0.8$ A is adopted for this interval, and it takes 31 ns to finish OFF–ON mode. After the EOS, the circuit is in Ohm-Off mode, where the primary switching transient has finished. A constant $I_g = 0.8$ A is also adopted for this interval. It can be inferred from the EOS that the predicted sequence length is changing for different cases, and the lengths of the three AGD sequences match the duration of the individual switching transients. The results demonstrate that the GRU-EDRNN can predict correct AGD sequences according to the switching targets.

In Cases 1 and 2, AGD reduces the maximum dv/dt to improve the common mode noise. The declines on the V_{CM} are evident. As quantified in Table 2, when the maximum dv/dt in Case 1 and 2 are decreased by 14.2% and 20.5%, the V_{CM} drops from -138 V to -123 V and -116 V, respectively. In Case 3, the focus is not dv/dt reduction, and the V_{CM} is unchanged compared to the CGD-1 case.

TABLE 3. Switching Transient Improvement by CGD and AGD

Reduction	Case	E_{sw}	di/dt	dv/dt
V_{CM}	CGD-1	0.3000	0.2572	0.4564
	CGD-1(Slow)	0.3350	0.2229	0.3979
	AGD-1	0.2738	0.2856	0.3915
I_{OV}	CGD-2	0.1875	0.5224	0.8077
	CGD-2(Slow)	0.2938	0.2681	0.4737
	AGD-4	0.2462	0.2470	0.5961

The average di/dt for Case 1 and 3 are increased by 11.0% and 37.4%, respectively, while for Case 2, it is kept the same as the CGD-1 case. The result is verified by the I_{ds} waveforms in Fig. 10(a). In Case 3, the V_{CM} is not improved, but the E_{sw} is minimized by significantly accelerating the di/dt interval. In Case 2, the V_{CM} is minimized at the cost of higher E_{sw}. Case 1 is when the optimum solution is found to improve both E_{sw} and V_{CM}.

It is worth noting that di/dt and dv/dt are controlled individually by the AGD, but for CGD-1, it is impracticable. Due to a high time resolution of 3 ns, the sequence predicted for the current rising and voltage falling intervals can fine-tune the switching transient. The AGD is thus an efficacious tool for switching transient improvement with the sequence predictor as a critical enabler.

B. SWITCHING TRANSIENT IMPROVEMENT: AGD VERSUS CGD

This study optimizes V_{CM} and turn-ON current overshoot I_{OV}. The AGD and CGD are applied to fulfill the targets, and the E_{sw} are compared. It shows that the AGD with the predicted driving sequence generates a lower loss in both cases to achieve the same improvement. The switching targets input to the GRU-EDRNN are the same as Section V-A since the maximum dv/dt affects V_{CM} and the average di/dt changes the I_{OV}. Table 3 summarizes the verification results.

For V_{CM} reduction, CGD-1(Slow) with higher R_g of 29 Ω is utilized to slow down the maximum dv/dt. It is discernable from the V_{CM} waveforms in Fig. 10(b) and Table 3 that CGD-1(Slow) and AGD-1 achieve similar V_{CM} values, but the AGD-1 achieves lower E_{sw} due to faster current transient speed, as shown in E_{sw} and I_{ds} waveforms. In the I_{OV} reduction case, CGD-2 with R_g of 12 Ω accomplishes low switching loss but introduces serious current overshoot, which can be seen from I_{ds} waveforms in Fig. 10(c). CGD-2(Slow) with higher R_g of 24 Ω, as a remedy, slows down the di/dt. The AGD-4 achieves the same overshoot reduction, but the switching loss is kept low due to acceleration on voltage transient speed, as inferred from E_{sw} and V_{ds} waveforms in Fig. 10(c). AGD outperforms CGD due to the augmented freedom on controlling di/dt and dv/dt. The GRU-EDRNN accurately generates AGD sequences for switching transient improvement based on the switching targets.

C. OPTIMALITY OF THE PREDICTION

The global optimality of the predicted sequence cannot be guaranteed since the problem of minimizing the loss and training a deep neural network is nonconvex [31], [32]. Therefore, the local optimality is investigated in this section. As an analogy to do partial derivatives for proving local optimality, a sequence perturbation method is adopted for the investigation.

The Case 3 discussed in Section V-A is taken as an example to explain the AGD sequence perturbation method. The predicted sequence is changed manually with small perturbations, and the modified sequence is applied to the verification circuit. The new switching results are then extracted and the average error is compared, which is defined as follows:

$$E_{rr} = \frac{1}{3}\left(\frac{|E_{sw} - E_{sw}^*|}{E_{sw}^*} + \frac{|di/dt - di/dt^*|}{di/dt^*} + \frac{|dv/dt - dv/dt^*|}{dv/dt^*}\right).$$

Table 4 tabulates the Case 3 AGD sequence and three perturbations.

1) PERTURBATION 1

The position for $I_g(3) = 1.6$ A is swapped with its adjacent elements, which changes the order of the gate driving sequence. The average error increases remarkably when the modified gate driving sequences are applied.

2) PERTURBATION 2

The strength of $I_g(3)$ is perturbed. The average error increases when the driving current is deviated from the prediction. As the deviation gets larger, the average error becomes higher.

3) PERTURBATION 3

The value of $I_g(2)$ to $I_g(4)$ are modified so that the new sequence injects the same total charge $Q_c = \sum_{i=2}^{4} I_g(i) \times 3$ ns. In this variation, the average error is considerably reduced compared to Variation 1 and 2. Nevertheless, the original predicted sequence still outperforms the two gate driving sequences in Perturbation 3.

The above results demonstrate that the GRU-EDRNN prediction achieves the lowest average error, while all the other variations underperform this neural network prediction. The discussion above is not exhaustive to prove the local optimality of the entire AGD sequence. Similar perturbations can be applied to other I_g predictions. Nevertheless, the predicted AGD sequence by the GRU-EDRNN best achieves the switching target among all the perturbed sequences; and hence, the local optimality of the prediction is partially validated.

TABLE 4. Perturbations to Case 3 AGD Sequence and the Corresponding Switching Results

Case	Sequence($\times 0.1A$)	E_{sw}	di/dt	dv/dt	Avg. Error
Case 3 target		E^*_{sw}=0.2380	di/dt^*=0.3530	dv/dt^*=0.4520	0.0%
Case 3 sequence	[5, 6, 6, 16, 6, 4, 4, 5, 4, 4, 5, 6, 6, 4, 6, 6, 6, 5]	0.2412	0.3534	0.4507	0.6%
Perturbation 1	[5, 6, **16**, **6**, **6**, 4, 4, 5, 4, 4, 5, 6, 6, 4, 6, 6, 6, 5]	0.2497	0.3455	0.4231	4.5%
(position swap)	[5, 6, **6**, **6**, **16**, 4, 4, 5, 4, 4, 5, 6, 6, 4, 6, 6, 6, 5]	0.2356	0.3558	0.5144	5.2%
Perturbation 2	[5, 6, 6, **15**, 6, 4, 4, 5, 4, 4, 5, 6, 6, 4, 6, 6, 6, 5]	0.2453	0.3385	0.4391	3.3%
(strength change)	[5, 6, 6, **17**, 6, 4, 4, 5, 4, 4, 5, 6, 6, 4, 6, 6, 6, 5]	0.2372	0.3702	0.4663	2.8%
	[5, 6, 6, **18**, 6, 4, 4, 5, 4, 4, 5, 6, 6, 4, 6, 6, 6, 5]	0.2331	0.3848	0.4859	6.2%
Perturbation 3	[5, 6, 6, **17**, **5**, 4, 4, 5, 4, 4, 5, 6, 6, 4, 6, 6, 6, 5]	0.2416	0.3543	0.4439	1.2%
(equal total charge)	[5, 6, **7**, **15**, 6, 4, 4, 5, 4, 4, 5, 6, 6, 4, 6, 6, 6, 5]	0.2421	0.3526	0.4487	0.9%

VI. CONCLUSION

The GRU-EDRNN proposed in this article provides one superior solution to address the AGD sequence prediction challenges. The superiorities are: 1) No search process is involved, and the sequence is predicted for given switching targets; and 2) the sequence length and individual driving strength are both generated by the GRU-EDRNN and are dynamically matched, which is a unique feature not seen in other approaches. The AGD sequence generation is the critical step of active gate driver development. With the proposed solution, the device switching transient parameters are optimized for switching loss (E_{sw}) reduction or EMI profile (related to di/dt and dv/dt) improvement.

ACKNOWLEDGMENT

The authors would like to thank Dr. Suxuan Guo for collaborating on the project. The presented work is part of the author's Ph.D. dissertation at NC State University [33].

REFERENCES

[1] Y. Yang, Y. Wen, and Y. Gao, "A novel active gate driver for improving switching performance of high-power SiC MOSFET modules," *IEEE Trans. Power Electron.*, vol. 34, no. 8, pp. 7775–7787, Aug. 2019.

[2] S. Zhao et al., "Adaptive multi-level active gate drivers for SiC power devices," *IEEE Trans. Power Electron.*, vol. 35, no. 2, pp. 1882–1898, Feb. 2020.

[3] A. Schindler, B. Koeppl, and B. Wicht, "EMC and switching loss improvement for fast switching power stages by di/dt, dv/dt optimization with 10 ns variable current source gate driver," in *Proc. 10th Int. Workshop Electromagn. Compat. Integr. Circuits*, 2015, pp. 18–23.

[4] W. Frank, A. Arens, and S. Hoerold, "Real-time adjustable gate current control IC solves dv/dt problems in electric drives," in *in Proc. Europe Int. Exhib. Conf. Power Electron., Intell. Motion, Renewable Energy Energy Manag.*, 2014, pp. 1–7.

[5] K. Yamaguchi, K. Katsura, T. Yamada, and Y. Sato, "Comprehensive evaluation of gate boost driver for SiC-MOSFETs," in *Proc. IEEE Energy Convers. Congr. Expo.*, Milwaukee, WI, 2016, pp. 1–8.

[6] P. Bau, M. Cousineau, B. Cougo, F. Richardeau, D. Colin, and N. Rouger, "A CMOS gate driver with ultra-fast dV/dt embedded control dedicated to optimum EMI and turn-on losses management for GaN power transistors," in *Proc. 14th Conf. Ph.D. Res. Microelectronics Electron.*, 2018, pp. 105–108.

[7] V. Krishna M. and K. Hatua, "Current controlled active gate driver for 1200 V SiC MOSFET," in *Proc. IEEE Int. Conf. Power Electron., Drives Energy Syst.*, 2016, pp. 1–6.

[8] H. C. P. Dymond et al., "A 6.7-GHz active gate driver for GaN FETs to combat overshoot, ringing, and EMI," *IEEE Trans. Power Electron.*, vol. 33, no. 1, pp. 581–594, Jan. 2018.

[9] K. Miyazaki et al., "General-purpose clocked gate driver IC with programmable 63-Level drivability to optimize overshoot and energy loss in switching by a simulated annealing algorithm," *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 2350–2357, May/Jun. 2017.

[10] J. Yu, W. J. Zhang, A. Shorten, R. Li, and W. T. Ng, "A smart gate driver IC for GaN power transistors," in *Proc. IEEE 30th Int. Symp. Power Semicond. Devices ICs*, 2018, pp. 84–87.

[11] S. Acharya, X. She, F. Tao, T. Frangieh, M. H. Todorovic, and R. Datta, "Active gate driver for SiC-MOSFET-Based PV inverter with enhanced operating range," *IEEE Trans. Ind. Appl.*, vol. 55, no. 2, pp. 1677–1689, Mar/Apr. 2019.

[12] P. Nayak and K. Hatua, "Parasitic inductance and capacitance-assisted active gate driving technique to minimize switching loss of SiC MOSFET," *IEEE Trans. Ind. Electron.*, vol. 64, no. 10, pp. 8288–8298, Oct. 2017.

[13] Y. S. Cheng, T. Mannen, K. Wada, K. Miyazaki, M. Takamiya, and T. Sakurai, "Optimization platform to find a switching pattern of digital active gate drive for reducing both switching loss and surge voltage," *IEEE Trans. Ind. Appl.*, vol. 55, no. 5, pp. 5023–5031, Sep./Oct. 2019.

[14] L. Middelstaedt, J. Wang, B. H. Stark, and A. Lindemann, "Direct approach of simultaneously eliminating EMI-Critical oscillations and decreasing switching losses for wide bandgap power semiconductors," *IEEE Trans. Power Electron.*, vol. 34, no. 11, pp. 10376–10380, Nov. 2019.

[15] "C2M0025120D silicon carbide power MOSFET," datasheet, Rev. B., 10-2015.

[16] R. Kraus and A. Castellazzi, "A physics-based compact model of SiC power MOSFETs," *IEEE Trans. Power Electron.*, vol. 31, no. 8, pp. 5863–5870, Aug. 2016.

[17] Y. Ren, M. Xu, J. Zhou, and F. C. Lee, "Analytical loss model of power MOSFET," *IEEE Trans. Power Electron.*, vol. 21, no. 2, pp. 310–319, Mar. 2006.

[18] S. K. Roy and K. Basu, "Analytical estimation of turn on switching loss of SiC mosfet and schottky diode pair from datasheet parameters," *IEEE Trans. Power Electron.*, vol. 34, no. 9, pp. 9118–9130, Sep. 2019.

[19] J. Wang, H. S. Chung, and R. T. Li, "Characterization and experimental assessment of the effects of parasitic elements on the MOSFET switching performance," *IEEE Trans. Power Electron.*, vol. 28, no. 1, pp. 573–590, Jan. 2013.

[20] R. Ahmed, R. Todd, and A. J. Forsyth, "Predicting SiC MOSFET behavior under hard-switching, soft-switching, and false turn-on conditions," *IEEE Trans. Ind. Electron.*, vol. 64, no. 11, pp. 9001–9011, Nov. 2017.

[21] Y. Xiong, S. Sun, H. Jia, P. Shea, and Z. John Shen, "New physical insights on power MOSFET switching losses," *IEEE Trans. Power Electron.*, vol. 24, no. 2, pp. 525–531, Feb. 2009.

[22] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. 2014 Deep Learn. Representation Learn. Workshop*, 2014, *arXiv:1412.3555v1*.
[23] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Empirical Methods Natural Lang. Process.*, .2014, *arXiv:1406.1078v3*.
[24] A. Karpathy and F. Li, "Deep visual-semantic alignments for generating image descriptions," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, *arXiv:1412.2306v2*.
[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–465.
[26] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, *arXiv:1805.11604*.
[27] A. Paszke et al.,, "Pytorch: An imperative style, high-performance deep learning library," in *Conf. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A. simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, 2014, vol. 15, no. 56, pp. 1929–1958.
[30] A. D. Brovont and A. N. Lemmon, "Common-mode/differential-mode interactions in asymmetric converter structures," in *Proc. IEEE Electric Ship Technol. Symp.*, 2017, pp. 84–90.
[31] B. D. Haeffele and R. Vidal, "Global optimality in neural network training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4390–4398.
[32] C. Yun, S. Sra, and A. Jadbabaie, "Global optimality conditions for deep neural networks," in *Proc. Int. Conf. Learn. Representations*, 2018, *arXiv:1707.02444v3*.
[33] L. Yang, "Design and control of an electrical vehicle traction inverter to address the opportunity and challenge of SiC wide bandgap device," Ph.D. dissertation, Dept. Elect. and Comp. Eng., NC State Univ., Raleigh, NC, USA, 2020. p. 111.

LI YANG received the M.S. degree from Xidian University, Xi'an, China, and the Ph.D. degree from the FREEDM System Center, North Carolina State University, Raleigh, NC, USA, both in electrical engineering in 2013 and 2020, respectively.

Since 2021, he is with Lucid Motors, Newark, CA, USA, developing charging products. His research interests include traction inverter design with wide-bandgap (WBG) devices, motor control, active gate driver, and WBG device modeling.

YUXUAN LIU received the B.S. degree in thermal energy and power engineering, and the M.S. degree in chemical engineering and technology, both from the Harbin Institute of Technology, Harbin, China, in 2016 and 2018, respectively, and the Ph.D. degree in mechanical engineering from North Carolina State University, Raleigh, NC, USA, in 2022.

Her research interests include materials mechanics, electrochemistry, microfabrication, and wearable electronics.

WENSONG YU (Member, IEEE) received the M.S. degree from the Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from South China University of Technology, China, both in mechanical and electrical engineering, in 1995 and 2000, respectively.

From 2006 to 2013, he was a Postdoctoral Researcher, Research Scientist, and Research Assistant Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA. Since 2013, he has been with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA, as a Research Associate Professor.

His current research interests include high-frequency solid-state transformer, advanced soft-switching technique, digital control of multiswitch topology, wide bandgap device applications, grid-forming inverters, high-voltage power conversion and protection, electric vehicle traction drive, distributed energy storage devices, and green energy grid infrastructure.

IQBAL HUSAIN (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 1993,

He is the Director of the FREEDM NSF Engineering Center and the ABB Distinguished Professor with North Carolina State University, Raleigh, NC, USA. Prior to joining NC State, he was with the University of Akron, Akron, OH, USA, where he developed the motor drives and electric vehicles program. He is the author of the textbook Electric and Hybrid Vehicles: Design Fundamentals (CRC Press, 2021). In 2022, he was the Visiting Faculty in several universities in Australia with a Fulbright Scholar Award working on renewable and electrified systems for green mining. He was also a Visiting Professor with Oregon State University, Corvallis, OR, USA, in 2001. His research interests include power electronic system integration into the power grid and transportation electrification with wide bandgap power electronic devices and permanent magnet and reluctance machine drives.

Dr. Husain was the recipient of the 2022 Alcoa Distinguished Engineering Research award at NC State, 2006 SAE Vincent Bendix Automotive Electronics Engineering award, the 2004 College of Engineering Outstanding Researcher award, the 2000 IEEE Third Millennium Medal, and the 1998 IEEE-IAS Outstanding Young Member award. He is the past Editor-in-Chief for *IEEE Electrification* Magazine.