Data-Driven Component-Level Design of Switching Power Converters Using Predictive Modeling

Skye Reese, *Student Member, IEEE*, Bailey Sauter, and Dragan Maksimović, *Fellow, IEEE*

***Abstract*—This article shows how the design of switching power converters at the component level can be assisted by data-driven, automated, systematic methods. This work uses regression machine learning (ML) techniques, where predictive component models are trained using large amounts of component data and design-oriented parameters derived from the data. For a given converter topology and specification set, the ML-based component models are used to predict converter power loss, bill-of-material (BOM) cost, and footprint area. This allows for an algorithm to iterate over design parameters to optimize the objective function in terms of efficiency, size, cost, or a combination of these performance metrics while also meeting all design constraints. Once the algorithm converges, the optimized parameters are applied to filter the original dataset, identifying components that are most likely to be chosen for the specific design. The developed design automation approach allows for comparisons between topologies, device technologies, and operating modes, and reduces the design time by quickly providing component recommendations. To illustrate an application of the ML-based design approach, the tool is applied to the design of 48-to-12 V, 5 A synchronous buck dc–dc converters using silicon or gallium nitride power semiconductors, where the design objective is to minimize loss while meeting BOM cost and area constraints.**

***Index Terms*—Converter optimization, data-driven component models, machine learning (ML), power converter design, switched-mode power converters.**

I. Introduction

A COMMON objective in designing switched-mode power converters is to meet the specifications while minimizing loss, size, cost, or a combination of these converter performance metrics. This objective is achieved via a design process typically consisting of many iterative design steps, as outlined in the flowchart of Fig. 1. The figure emphasizes the scope of the approach presented in this article, which focuses on streamlining component selection to make this step more systematic and efficient, thus reducing the need for manual design iterations.

Consider, as an example, the design of a simple synchronous buck converter shown in Fig. 2. Given the specifications, the design objectives, and the converter steady-state solution, operating characteristics, and loss models, the designer may proceed to manually select the power stage components among many thousands of available options, the switching frequency, and the operating mode, e.g., hard-switched continuous conduction mode, discontinuous conduction mode, or soft-switched zero-voltage-switching quasi-square-wave mode. Considering just the selection of the transistors Q_1, Q_2, the inductor L_1, and the capacitors C_{in}, C_{out}, after taking an initial Pareto front of a set of collected commercially available components, there are 830, 1100, and 18 000 possible transistors, inductors, and capacitors, respectively. If all components were considered, the design space would consist of approximately 10^{17} combinations. The design

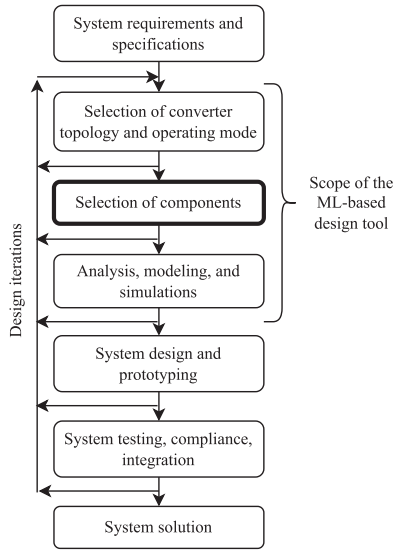

Fig. 1. Flowchart showing where the presented design approach fits into the overall power electronics design process. Automating the component selection step contributes to the overall goal of automated converter design assistance.

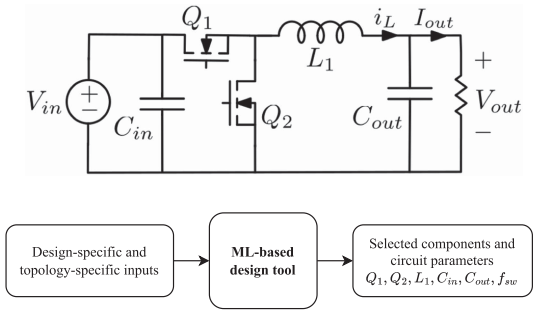

Fig. 2. Top-level flowchart of how the ML-based design tool is used to select components in a synchronous buck converter example.

Received 17 September 2024; revised 28 January 2025; accepted 12 March 2025. Date of publication 26 March 2025; date of current version 26 May 2025. This work was supported in part by the National Science Foundation GRFP under Grant 1000315471. Recommended for publication by Associate Editor H. S.-H. Chung. *(Corresponding author: Skye Reese.)*

The authors are with the Colorado Power Electronics Center, Department of Electrical, Computer and Energy Engineering, University of Colorado, Boulder, CO 80309 USA (e-mail: skre2597@outlook.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPEL.2025.3555147.

Digital Object Identifier 10.1109/TPEL.2025.3555147

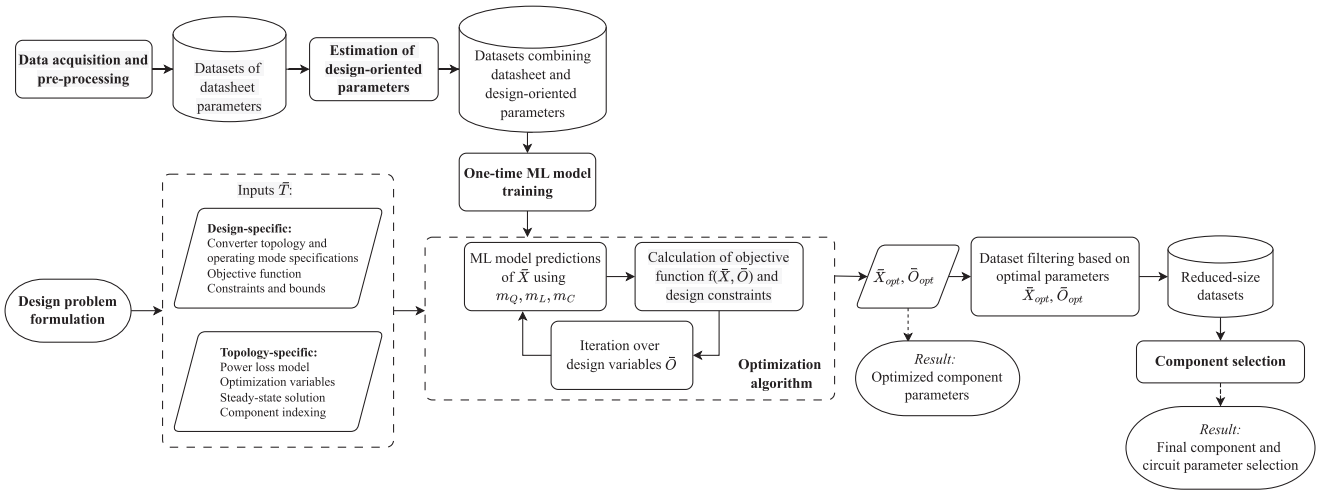

Fig. 3. Flowchart of the converter design tool using machine learning-based predictive component modeling. Each highlighted item is discussed in detail in a corresponding section: data acquisition and preprocessing in Section III, design-oriented parameters in Section IV, ML modeling in Section V, optimization in Section II, component selection in Section VI, and the design problem formulation and implementation for buck converter examples in Section VII.

challenges become even more pronounced in power electronics systems that involve more complex converter topologies and tradeoffs in converter architectures and operating modes, performance metrics, and modulation strategies [1].

Given that an exhaustive search is impractical, a common heuristic approach involves the designer manually narrowing the design space, guided by domain expertise and prior experience. Such standard design approaches can be improved by incorporating more systematic data-driven approaches, and such design automation methods have been gaining attention [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21].

Data-driven heuristic design automation methods reported in power electronics literature address the following three key issues:

1) data acquisition;
2) the application of machine learning (ML) tools;
3) optimization over large design spaces.

Empirical data acquisition has been limited in scope to modeling aspects where experimental setups for systematic characterization are feasible, such as core loss modeling [4], [5], [6]. For a broader design context, component or converter-level data can be collected using analytical modeling and simulation methods [7], [8], [9], [10], as well as automated extraction from datasheets [11], [12].

Many of the data-driven methods currently proposed in literature incorporate ML models to capture trends in the presented datasets. For example, the authors in [7] and [8] used reinforcement learning, whereas Li et al. [9] used a multilayered batch-normalization artificial neural network (ANN) to deduce important features and make predictions about the power loss of a given design. An ANN combined with transfer learning is used in [10] to determine the frequency, inductance, and capacitance to optimize efficiency, while Lin et al. [1] used a physics-in-architecture recurrent neural network to select optimal operating conditions and structure for a three-level NPC dual-active-bridge converter.

The next stage common across data-driven methods is an optimization structure to solve the combinatorial optimization problem in terms of component parameters or operating conditions. For example, Li et al. [9] used a genetic algorithm (GA) to search across the design space and make predictions about power loss at each iteration, and then employs look-up tables to identify specific components. Deep deterministic policy gradient, a policy-based learning process, is used in [8] to determine optimal parameters for an optimized switching frequency, while [10] uses particle swarm optimization. A multiobjective combinatorial optimization algorithm is used in [13] to find optimal magnetic and capacitive component parameters.

The data-driven approach [18] and the design tool presented in this article and depicted in the block diagram of Fig. 3 differ from the previously proposed approaches in all three considered directions, including the following:

1) data acquisition;
2) the structure and the application of ML models;
3) the way an optimization step is implemented.

First, comprehensive component datasets are assembled by systematically scraping distributor sites and component datasheets. Compared to previously reported approaches, the datasets are based on *real, commercially available components*, and include pricing and packaging data. Furthermore, the approach extends the component datasets with *design-oriented parameters* necessary for converter loss modeling [19]. These design-oriented parameters are derived in an automated manner using predictive ML modeling. Second, trends among component parameters are captured using relatively simple *regression ML models*. Unlike deep learning methods, the selected regression ML models are easy to train on moderately sized datasets and require minimal computational effort to generate predictions. This efficiency is particularly important, as ML-based component models are frequently invoked within the optimization step. Notably, the one-time trained component models can be reused across the design of various converter topologies. This contrasts with many previously reported methods, where ML

models are tailored to specific topologies and must be retrained for each new topology or design problem. Finally, inspired by the methods discussed in [20] and [22], the developed ML models enable *continuous optimization* over the large design space, leading to inherently fast and robust convergence. The continuous optimization also implies that the results reveal valuable insights into the impact of component parameters, and are less sensitive to specific component availability. In the final step of the proposed design process, component sets are provided using real, commercially available part numbers, aiding the designer in achieving the design objectives. To facilitate the use and further developments, the code repository for all the developed processes and methods has been published [23].

The rest of this article is organized as follows. Section II describes the overall structure of the ML-based design tool, the block diagram of which is shown in Fig. 3. Section III discusses gathering and preprocessing the data on transistors, inductors, and capacitors. Section IV details the development of design-oriented parameters that combine datasheet data with device physics in order to represent the main loss mechanisms that contribute to converter power losses. Section V describes how the ML models are constructed to capture trends among component parameters, and how these models are used to predict the quantities needed in the converter optimization step. Limitations of the ML-based component models are also discussed. Section VI details the final step of the approach, where the ML-based results are used to develop small subsets of components. These subsets are used within a search algorithm that leads to final component selection. Section VII tests the design tool on 48-to-12 V, 5 A dc–dc buck converter examples using silicon (Si) or gallium nitride (GaN) power semiconductors, and includes experimental validation using the components suggested by the design tool. Finally, Section VIII concludes this article.

II. ML-Based Converter Design Approach

The block diagram in Fig. 3 visualizes the design approach developed in this article. The scraped component data (see Section III) is combined with derived design-oriented parameters (see Section IV) and used to train ML models (see Section V), which use design variables $\bar{O}$ and inputs $\bar{T}$ to predict the component parameters $\bar{X}$ necessary to evaluate loss $P_{\text{loss,total}}$, BOM cost U_{tot}, and BOM footprint area A_{tot}. For each transistor, inductor, and capacitor included in a design, the parameter sets for $\bar{O}$, $\bar{T}$, and $\bar{X}$ are, respectively, the same (as shown in Table V of Section V), which implies that the one-time trained ML models are generally applicable to the design of different converter topologies. Furthermore, the ML models can represent the relationships between component parameters as continuous functions, capturing underlying trends and patterns. Consequently, the combinatorial optimization problem is transformed into a continuous one.

At each iteration, the selected optimization algorithm uses $\bar{O}$ and $\bar{X}$ to compute the objective function $f(\bar{X}, \bar{O})$ and check the constraints, then iterates over the design variables $\bar{O}$. When the algorithm converges, the predicted theoretical optimal parameters $\bar{X}_{\text{opt}}$ are used to reduce the component dataset to a size easily handled by a search algorithm (see Section VI), which returns the recommended component part numbers and operating conditions.

A. Optimization Problem Formulation

The converter design is posed as a constrained optimization problem

$$\begin{aligned} &\min_{\bar{O}} f(\bar{X}, \bar{O}) \\ &U_{\text{tot}} \leq U_{\text{max,total}} \\ &A_{\text{tot}} \leq A_{\text{max,total}}. \end{aligned} \tag{1}$$

The objective is to minimize a user-defined converter performance metric $f(\bar{X}, \bar{O})$, such as power loss $P_{\text{loss,total}}$, under user-defined constraints, such as BOM cost U and BOM area A limits. The objective function $f(\bar{X}, \bar{O})$ and the constraints can be defined in many different ways, depending on the user's specified design objectives. In the example presented in Section VII, the objective function is selected to be the total loss $P_{\text{loss,total}}$. Alternatively, the objective function may, for example, be defined as a product or a normalized sum of $P_{\text{loss,total}}$, U_{tot}, and/or A_{tot}.

Input $\bar{T}$ is split up into inputs related to the overall converter $\bar{T}_{\text{con}}$ and inputs related to each device $\bar{T}_Q, \bar{T}_L, \bar{T}_C$ for transistors, inductors, and capacitors, respectively. The mapping from $\bar{O}, \bar{T}$, to $\bar{X}$ required to compute $f(\bar{X}, \bar{O})$ is facilitated by the one-time trained ML-based component models

$$\bar{O}, \bar{T} \underset{\text{ML models}}{\longrightarrow} \bar{X}. \tag{2}$$

The component ML models m_Q, m_L, m_C predict $\bar{X}_Q, \bar{X}_L, \bar{X}_C$, which are all the parameters required for computing the power loss, cost, and area-related quantities of each component. A flowchart of how the ML models are structured and incorporated into the design approach is shown in Fig. 4. The output of the optimization algorithm $\bar{X}_{\text{opt}}, \bar{O}_{\text{opt}}$ is used to filter each component dataset, essentially serving as a secondary Pareto-front standard that takes into account the specific design, topology, and design objective. These subsets of components are used in an exhaustive search across component combinations, as discussed in Section VI.

B. Optimization Algorithm

By reducing the optimization problem to a continuous form, a variety of well-established and widely-used continuous optimization algorithms can now be integrated into the flowchart of Fig. 3. The selected algorithm is constrained optimization by linear approximation (COBYLA) [24], which is a constrained, bounded, multivariate optimization method, an implementation of which is available via the Scipy optimize class [25]. COBYLA creates successive linear approximations of both the objective function and constraints. At each iteration, the algorithm minimizes the linear approximation of the objective function and constraints within a trust region. The resulting candidate solution is then evaluated against the original objective function and constraints. The trust region is progressively reduced to refine

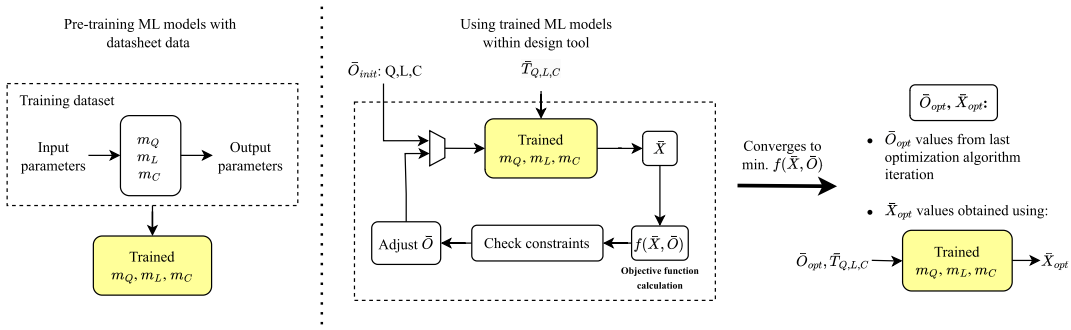

Fig. 4. Flowchart showing a zoomed-in version of how the pretrained ML models are used in the design tool.

the search, with convergence achieved when the trust region size falls below a specified threshold.

The algorithm hyperparameters are chosen to assist in successful convergence and effectively search over the entire design space. The number of iterations was set to 1000, as convergence failure beyond 1000 iterations typically indicated overly restrictive constraints, rendering the design unfeasible. Successful termination typically occurs within a few minutes on a standard desktop. The termination threshold was set to balance escaping local minima and maintaining precision near the optimum. Constraint violation tolerance was set near zero, to reflect strict design requirements. Future work could improve the handling of constraint violations by accounting for uncertainties in ML model predictions.

C. Design Tool Implementation

The design tool diagrammed in Fig. 3 and implemented in Python is built so that it can be applied to any converter topology and any set of design specifications, and the corresponding code is open-sourced [23]. The inputs $\bar{T}$ are broken into two groups: design- and topology-specific.

The first topology-specific input is the set of component indices to be included in the optimization process. The designer can choose to include any number of transistors (Q), inductors (L), and capacitors (C). The second topology-specific input is the set of optimization variables, where the designer determines which variables the tool should iterate over. The third topology-specific input is the set of steady-state solution equations, including standard relationships between control variables, such as the duty cycle and switching frequency, and operating characteristics, such as the conversion ratio, voltage and current stresses, and switching ripples. The fourth topology-specific input is a power loss model. This loss model consists of two parts: the computation of relevant datasheet-based and physics-based loss parameters for each component, and the equations that compute the total power loss based on the precomputed physics-based loss parameters and the operating conditions. The pretrained ML models, the optimization algorithm, and the final step in which specific component candidates are selected remain the same regardless of the specific converter topology or design-specific inputs.

The design-related inputs include the selected converter topology label, the operating mode selection, and the design specifications: input and output requirements, any component rating and value constraints, the selected converter metric to be optimized, and the design constraints, e.g., on power efficiency, BOM cost, or BOM footprint area.

D. Results

The tool produces results of interest at two stages of the ML-based design process, as shown in Fig. 3. First, the optimization step outputs $\bar{X}_{\text{opt}}$, $\bar{O}_{\text{opt}}$, offer quick performance estimates and enable evaluations of the impact of component parameters, as well as comparisons of different topologies, operating conditions, and constraints without the need to consider specific components. The generated optimized component parameters $\bar{X}_{\text{opt}}, \bar{O}_{\text{opt}}$, can also be used as the initial values for the exhaustive search step regardless of changes in component availability. Second, the component selection step identifies and ranks viable component combinations that meet design constraints and optimize the objective function. Designers can choose from these combinations based on the performance ranking and additional factors such as up-to-date availability, pricing, or manufacturer preferences. The selected component combination can then be used to evaluate expected performance metrics including a detailed breakdown of power loss, cost, and area contributions, as shown in the examples provided in Section VII.

III. Data Acquisition and Preprocessing

The first step to developing the ML-based optimization tool is generating high-quality, high-quantity sets of training data for the ML-based component models. This section describes the process for obtaining and cleaning the datasets of power transistors, fixed inductors, and Class II multilayer ceramic (MLC) capacitors via outlier removal and Pareto dominance.

Using web-scraping tools available in Python [26], the component information on 22 000 power FETs, 48 000 fixed inductors, and 110 000 Class II MLC capacitors were gathered from commercial distributor sites, including DigiKey [27] and Mouser [28]. Some important device parameters and characteristics are only available in component datasheets, which vary in format and are not as easily accessible electronically. Examples include MOSFET body-diode reverse recovery charge (Q_{rr}) and device capacitance values, both important for modeling switching losses. To address this, a Python PDF-reader tool (Beautiful Soup [29]) was used to scrape PDF datasheets, resulting in a comprehensive dataset for 3100 transistors. A regular expression-based method was developed to extract specific

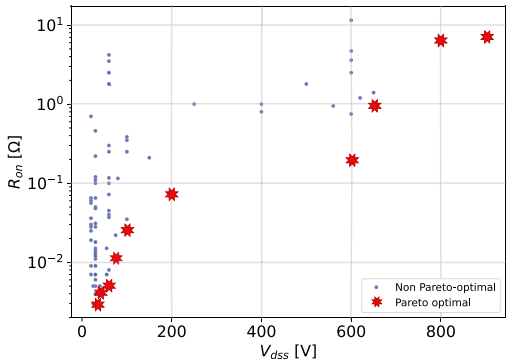

Fig. 5. Example of retaining the lowest envelope of data using the Pareto front in 2-D, where the only considered features are V_{dss} and R_{on}.

parameters from the scraped PDF data. This process was challenging due to inconsistent formatting among manufacturers, where labels such as Q_{rr} appeared as variations such as "Q RR," "Qrr," "Q rr," "Qr," or "Q_rr," and the spacing between labels and corresponding values differed significantly.

After gathering the data, various preprocessing techniques are applied to improve the quality of the datasets and retain the most relevant components. These techniques include outlier removal and Pareto dominance.

A. Outlier Removal

Visualizations of the data show that some components appear to be outliers with characteristics substantially different than the majority of similar components in the dataset. Upon manually investigating these individual outlier components, it is found that the parameter values reported on a distributor main webpage (and thus, what was scraped) occasionally differ from the values in the component datasheets. Various ML-based outlier removal techniques are available to automate the outlier identification and removal process. Here, we use a multivariate outlier removal method based on the Mahalanobis distance [30], where each datapoint is considered with respect to the trend. Cleaning the data in this way improves the predictive accuracy of ML models because model algorithms can be adversely influenced by outliers.

B. Pareto Dominance

Pareto dominance is a technique useful for determining the best envelope of components based on parameters, i.e., dimensions of interest [31]. Pareto dominance applied to converter design looks at all of the specified features for a dataset and determines if a component is strictly worse than the rest and, therefore, unlikely to be used in an optimized design [18]. Taking the Pareto front on n dimensions yields an $(n-1)$-dimensional surface of Pareto optimal data points. This is most clearly seen when taking the Pareto front on two dimensions (e.g., voltage rating V_{dss} and on-resistance R_{on}), as shown in Fig. 5 for silicon (Si) power MOSFETs. The features included in taking the Pareto front for transistors, inductors, and capacitors are shown in Table I.

TABLE I
FEATURES INCLUDED IN TAKING THE PARETO FRONT OF EACH COMPONENT'S DATASET

	Features
Transistors	V_{dss}, R_{on}, Q_g, U, A
Inductors	I_{max}, R_{dc}, U, A, H
Capacitors	V_{rated}, U, A, H

U = Component cost, A = Package area, and H = Package height.

IV. Estimation of Design-Oriented Parameters

A method of representing the losses found within a converter is important in the optimization algorithm step in Fig. 3 to determine component parameters such that the losses can be minimized. However, this is a challenging problem because the device physics and loss mechanisms are complex and the relevant datasheet information is limited.

Converter loss models based on device datasheet parameters have been extensively investigated in the literature [32], [33], [34], [35], [36], [37], [38]. These models involve tradeoffs between accuracy, complexity, and the need to perform additional physics-based or empirical modeling steps to augment the datasheet data. An example of such a loss model using the datasheet and design-oriented parameters is provided for the synchronous buck converter in Appendix, Table XI. One may observe that even relatively simple loss models require parameters not directly available in device data on distributor sites or device datasheets.

This section discusses novel physics- and data-based approaches to the estimation of several design-oriented parameters, aiming to leverage predictive modeling to quantify the primary loss mechanisms associated with transistors, capacitors, and inductors.

A. R_{on} Versus T_j Dependence

Transistor on-resistances are reported on the distributor sites, but usually only at an operating temperature of $T_j = 25\,^{\circ}\text{C}$. To better represent practical operating conditions, where the transistor typically reaches an elevated temperature of, for example, $T_j = 100\,^{\circ}\text{C}$, this section aims to determine a multiplier k_T

$$k_T = \frac{R_{\text{on}}(T_j = 100\,^{\circ}\text{C})}{R_{\text{on}}(T_j = 25\,^{\circ}\text{C})}. \tag{3}$$

The transistors are split into five groupings; low-voltage ($\leq 200\,\text{V}$) and high-voltage ($> 200\,\text{V}$) Si, low-voltage ($\leq 100\,\text{V}$) and high-voltage ($> 100\,\text{V}$) GaN, and silicon carbide (SiC). These groupings were determined by examination of the k_T versus V_{dss} plots of each device type, which indicate slope changes at 200 V and 100 V for Si and GaN, respectively. The R_{on} versus T_j plots included in the datasheets of $\sim$25 components for each grouping were manually digitized. The components were randomly selected but intentionally spanned a range of voltage ratings and device manufacturers. From the manually digitized graphs, the multiplier k_T at $100\,^{\circ}\text{C}$ was recorded.

Table II shows a summary of the linear relationships and datasheet-validated performance results of the k_T estimation.

TABLE II
R_{ON} TEMPERATURE MULTIPLIER $k_T(V_{DSS})$ FOR EACH OF THE FIVE TRANSISTOR GROUPINGS, DETERMINED BY THE LINEAR BEST-FIT LINE OF THE k_T VERSUS V_{DSS} [V] DATA BASED ON MANUALLY DIGITIZED $R_{ON}(T_j)$ PLOTS FOR A RANDOM SELECTION OF COMPONENTS IN EACH GROUPING

Grouping	$k_T(V_{dss})$	MAPE [%]
Low-voltage Si	$0.233\log_{10} V_{dss} + 1.15$	9
High-voltage Si	$0.353\log_{10} V_{dss} + 0.83$	6
Low-voltage GaN	$0.066\log_{10} V_{dss} + 1.34$	3
High-voltage GaN	$0.148\log_{10} V_{dss} + 1.15$	4
SiC	$0.572\log_{10} V_{dss} - 0.523$	17

Performance results are based on the mean absolute percentage error (MAPE) [39] of $\sim$5 transistors that were hidden during training, comparing the estimated k_T values with the respective datasheet-calculated k_T values. A cross-validation technique was used for MAPE scoring, where data are set aside for training and testing in a cyclical manner and the average score of all iterations is returned.

B. Body Diode Reverse-Recovery Parameters

In a hard-switched converter, losses due to diode or body-diode reverse recovery can be very significant. This switching loss P_{Qrr} is modeled using the reverse recovery charge Q_{rr} and the reverse recovery time t_{rr}. Datasheets usually report only one set of Q_{rr}, t_{rr} values at a specific operating point and under specific test conditions characterized by the diode current I_F prior to turn-OFF, and the slope $\frac{di_F}{dt}$ of current decay. As summarized in Appendix, a device physics-based approach discussed in [34] and [40] can be used to determine the device time constants τ_c and τ_{rr}, which then allow one to estimate Q_{rr}, t_{rr}, and P_{Qrr} under the operating conditions considered in the design.

C. Charge-Equivalent Capacitance

In a hard-switched half-bridge stage, the energy loss due to the device output capacitance C_{oss} can be found using the charge-equivalent capacitance $C_{ds,Q}$ given by

$$C_{ds,Q} = \frac{1}{V_{off}} \int_0^{V_{off}} C_{oss}(V_{ds})\, dV_{ds} \tag{4}$$

where V_{off} is the voltage to which the device is exposed during the circuit off-state [37], [38], [41]. As shown in (4), $C_{ds,Q}$ requires C_{oss} as a function of drain-to-source voltage V_{ds}. MOSFET datasheets typically include $C_{oss}(V_{ds})$ in graphical form, as well as C_{oss} measured at a particular voltage $V_{ds,meas}$. Unfortunately, $V_{ds,meas}$ varies drastically between datasheets, implying inconsistent C_{oss} comparisons. For any given device, $C_{ds,Q}$ can be found by digitizing the $C_{oss}(V_{ds})$ graph, but it is difficult to automate this process on large MOSFET datasets. Due to this limitation, the approach taken here aims to approximate the $C_{oss}(V_{ds})$ curve for all components by combining the machine-scraped single reported values (C_{oss}, $V_{ds,meas}$) with trends determined by manual examination of select $C_{oss}(V_{dss})$ curves.

The MOSFETs are once again divided into five categories: low-voltage Si, high-voltage Si, SiC, low-voltage GaN, and high-voltage GaN. For each group, the C_{oss} versus V_{ds} plots

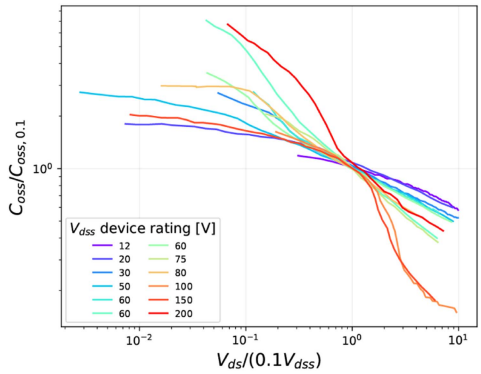

Fig. 6. Normalized C_{oss} versus V_{ds} curves for a randomly selected assortment of low voltage, Si MOSFETs from a variety of manufacturers. The V_{ds} values are normalized by $0.1V_{dss}$, and the C_{oss} values are normalized by $C_{oss,0.1} = C_{oss}(0.1V_{dss})$.

TABLE III
$\gamma(V_{DSS})$ FOR EACH OF THE FIVE GROUPINGS OF TRANSISTORS, ALONG WITH THE RESPECTIVE MAPE COMPARING THE PREDICTED γ WITH THE γ MANUALLY OBTAINED FROM THE DATASHEET FOR A SELECTION OF OUT-OF-SAMPLE COMPONENTS

Grouping	$\gamma(V_{dss})$	MAPE [%]
Low-voltage Si	$0.0021\, V_{dss} + 0.251$	11
High-voltage Si	$0.00089\, V_{dss} + 0.427$	33
Low-voltage GaN	$0.00062\, V_{dss} + 0.355$	8
High-voltage GaN	$0.00039\, V_{dss} + 0.353$	20
SiC	0.451	15

Note that the slopes γ for SiC components can be well-approximated by a constant mean value.

for a select number of components with varying voltage ratings and device manufacturers are collected and digitized into a machine-readable format. The plots confirmed that the same V_{dss} ranges are appropriate for Si and GaN devices as those used for k_T. These datapoints are then normalized by $0.1V_{dss}$ and $C_{oss,0.1} = C_{oss}(0.1V_{dss})$ as found from the digitized C_{oss} versus V_{ds} curve. As an example, Fig. 6 shows the normalized C_{oss} plots collected for low-voltage Si components, revealing an approximately linear trend on the log–log scales

$$C_{oss} \approx C_{oss,0.1} \left(\frac{0.1V_{dss}}{V_{ds}} \right)^{\gamma}. \tag{5}$$

By plotting the slope γ as a function of V_{dss}, the approximate linear fit for $\gamma(V_{dss})$ can be determined

$$\gamma = \gamma_1 V_{dss} + \gamma_0\,. \tag{6}$$

Given (6), V_{dss} of the component, and the machine-scraped (C_{oss}, $V_{ds,meas}$) values, (5) can be used to determine $C_{oss,0.1}$ and then calculate the charge equivalent capacitance using (4), without the need to digitize all C_{oss} versus V_{ds} plots.

Similarly to the validation approach described in Section IV-A, components not used to generate the $\gamma(V_{dss})$ equations are selected from each of the five transistor groups. First, the V_{dss} of the component and the corresponding $\gamma(V_{dss})$, as shown in Table III, are used to calculate γ. In comparison, the true γ is determined by finding the slope of the manually digitized C_{oss} versus V_{ds} plot reported in the datasheet. The

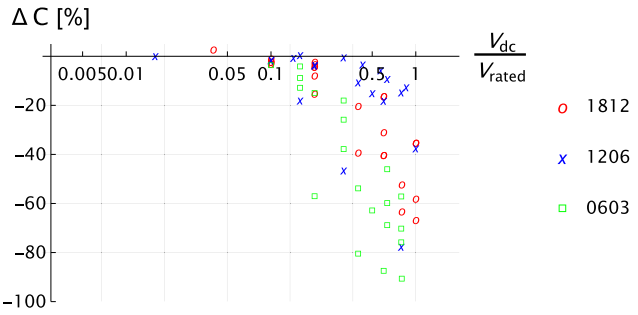

Fig. 7. ΔC versus $\frac{V_{dc}}{V_{rated}}$ for a variety of Class II MLC capacitors.

MAPE values between the predicted and reported values of γ are shown in Table III for each of the component groups, showing that (5) and (6) offer a good prediction of $C_{oss}(V_{ds})$ and thereby of $C_{ds,Q}$ in (4).

D. Bias Dependence of Class II Ceramic Capacitors

For Class II MLC capacitors, capacitance depends strongly on the applied dc bias voltage V_{dc} because of the effects of the change in permittivity of the material. Although ΔC versus V_{dc} graphs are available in a variety of formats, these are extremely difficult to gather in large quantities in an automated manner. The process is even more difficult because of the inconsistency with where such information is provided, such as manufacturer datasheets, online tools, or separate characterization sheets.

In this work, data-based techniques are used to automate the capacitance versus dc bias characterization across the dataset of Class II MLC capacitors. Using online graph-to-data digitization tools, the graph data of ΔC versus V_{dc} were collected for 50 Class II MLC capacitors from a variety of manufacturers, case sizes, voltage ratings, and capacitance.

An approximately linear relationship can be observed on the log-linear plot of ΔC versus $\frac{V_{dc}}{V_{rated}}$ in Fig. 7. These datapoints were used to train a linear regression ML model to find the expected capacitance C_{0Vdc} at zero dc bias. The known quantities in the design are V_{dc}, which is determined by the voltage seen across the capacitor, and C_{Vdc}, which can be calculated using the standard set of steady-state equations. C_{Vdc} and V_{dc} are used along with scraped quantities V_{rated}, A as inputs to predict output C_{0Vdc}. The MAPE was found to be 14.5%. The predicted output value of C_{0Vdc} is considered the minimum nominal capacitance. C_{0Vdc} is the quantity scraped via automation and can then be used as an input for predictions of other capacitor parameters such as cost U.

E. Inductor AC Loss Parameters

Datasheet parameters for fixed inductors typically include the nominal inductance L, inductance at saturation current L_{sat} @ I_{sat}, dc resistance R_{dc}, rms current rating at certain temperature rise I_{rms} @ ΔT, geometrical parameters of the package, and a core-material designation. Datasheets may also include the Q-factor at a particular frequency Q @ f_Q and the self-resonant frequency f_{res}. Unfortunately, the datasheet parameters are insufficient to accurately predict ac winding or core losses when inductors are used in switched-mode converters.

The goal of the approach presented in this section is to estimate the parameters required for the calculation of ac winding losses and core losses in off-the-shelf inductors, as summarized in Appendix. The presented approach is restricted to commercially available surface-mounted metal-powder-core fixed inductors.

Starting from the geometric parameters of a component, the ac loss parameter estimation algorithm iterates over the number of turns N_{turns}, adjusting the wire cross-sectional area to meet the datasheet value of R_{dc}. Next, the algorithm estimates the number of winding layers to fit the winding into the available window area and the relative permeability of the material necessary to obtain the datasheet value of inductance L. In each iteration, the Dowell's method [42, Sec. 10.4] is used to calculate the Q-factor of the inductor. Assuming approximately equal contributions of winding loss and core loss to the inductor Q-factor, iteration over N_{turns} stops when the estimated Q is closest to the datasheet value. Unfortunately, for many inductors, the Q-factor is not reported in the datasheets. This challenge is addressed by a simple linear correlation using the core volume V_c of the component

$$Q@100\,\text{kHz} \approx 0.0052\ V_c[\,\text{mm}^3] + 12. \tag{7}$$

The correlation (7) is based on data collected from a subset of inductors where Q is reported. Larger inductors generally having a larger Q-factor is consistent with the analysis in [43] and the data-based results reported in [44].

Once the number of turns N_{turns}, the wire size, and the number of layers are determined, the estimated R_{ac}/R_{dc} follows using the Dowell's method. To further simplify loss modeling, a piecewise fit on the log–log scales

$$R_{ac}(f)/R_{dc} \approx \max\left(1, (f/f_b)^b\right) \tag{8}$$

is adopted so that ac winding losses can be found for an arbitrary inductor current waveshape using only two design-oriented parameters: the break frequency f_b and the exponent b. Finally, the Steinmetz core-loss parameters K, α, and β are adopted from [45] for the metal powder material with μ_r closest to the estimated value for use in core loss estimation.

To validate the approach, the ac loss parameter estimation method has been automated and applied to a selection of 14 surface-mount powder-core fixed inductors from various manufacturers. The winding-loss results are summarized in Table IV, which uses the loss model equations from Table XI in Appendix: $P_{Lw,dc}$ using R_{dc}, P_{Lw} using the PWL fit (8) for R_{ac}, and the losses $P_{Lw,m}$ obtained with R_{ac}/R_{dc} measured experimentally using a Bode 100 network analyzer.

As a case study, the winding losses are calculated for a representative inductor current waveform that has a dc component $I_{dc} = I_{sat}/2$ and a triangular waveform with 100% ripple magnitude $\Delta i = I_{sat}/2$ at a switching frequency f_{sw}, where

$$f_{sw} = \max(5f_b, 20\,\text{kHz}) \tag{9}$$

and f_b is the frequency at which the measured R_{ac}/R_{dc} exceeds 1.1.

TABLE IV
RESULTS OF R_{AC} ESTIMATION FOR SELECTED FIXED INDUCTORS

Inductor index	L [µH]	I_{rms} [A]	$P_{Lw,dc}$ [W]	$P_{Lw,m}$ [W]	P_{Lw} [W]
1	2.2	12.0	1.58	2.13	1.78
2	5.6	12.5	1.50	1.89	2.32
3	10.0	5.6	1.77	2.47	3.00
4	10.0	7.0	2.54	3.04	2.71
5	4.7	1.2	0.10	0.12	0.10
6	1.0	11.0	0.10	0.16	0.16
7	6.8	5.5	0.24	0.34	0.39
8	1.2	30.0	1.61	1.64	2.78
9	4.7	27.0	0.79	1.15	1.68
10	2.2	7.1	0.29	0.33	0.30
11	1.0	10.1	0.25	0.45	0.45
12	10.0	30.0	0.51	1.05	1.71
13	1.0	23.0	2.27	2.30	3.44
14	1.0	43.5	1.01	3.98	1.31

The inductor part numbers by index are as follows:
(1: SRP1038A-2R2M, 2: SRP1265A-5R6M, 3: SRP8540A-100M,
4: 74437368100, 5: 744766904, 6: SRP4020FA-1R0M,
7: IHLP2525CZER6R8M11, 8: LMLP13B3M1R2DTAS, 9: HCMA1707-4R7-R,
10: DFEH7030D-2R2M=P3, 11: HCM0503-1R0-R, 12: SRP2313AA-100M,
13: 744373965010, 14: XAL1010-102MED).

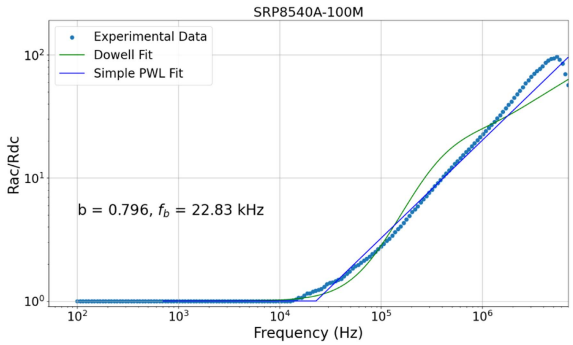

Fig. 8. R_{ac}/R_{dc} estimation results for SRP8540A-100M inductor, compared to Bode 100 measurements.

Fig. 8 illustrates the estimation results verified by R_{ac} measurements using the Bode 100 impedance analyzer for an example inductor where the predicted fit closely matches the measurements. The results in Table IV show that the basic model $P_{Lw,dc}$ always underestimates the winding loss. Despite some notable errors compared to $P_{Lw,m}$, which can be attributed to various assumptions the estimation algorithm must make, the improved estimate P_{Lw} mitigates the underestimation and provides a more accurate representation of the inductor winding loss.

V. ML MODELS

ML models can be used to capture trends among large quantities of component data [18], [44]. Fig. 3 shows how after one-time training, ML-based component models m_Q, m_L, m_C are used within the optimization loop to quickly estimate the component parameters needed to calculate converter power loss, BOM cost and footprint area. In this work, supervised learning-based ML models are developed, utilizing regression tools from [46]. Regression models offer a straightforward and efficient training and prediction process, making them a suitable choice for application in the design tool of Fig. 3, where the ML-based component models are invoked frequently within the optimization step.

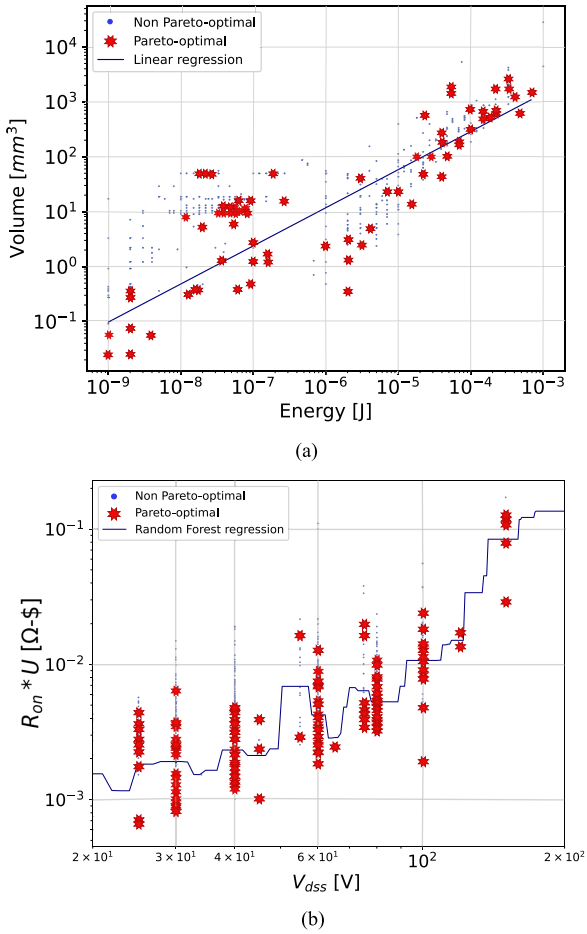

Fig. 9. Examples of datasets and ML regression models trained on Pareto dominant data points for (a) inductors with a fixed I_{max} current rating using only energy as an input, and (b) Si MOSFETs with a fixed R_{on} rating using only V_{dss} as an input.

Section V-A discusses how the ML models are structured, Section V-B summarizes the selected models, while Section V-C concludes with a discussion about the limitations of the ML models.

A. ML Model Structure

The preprocessed datasets[1] acquired as described in Section III are used to train the ML models. As an illustration, Fig. 9 shows 2-D plots of the datapoints after taking the Pareto front on all dimensions shown in Table I for inductor and transistor data. The data are overlaid by a linear and a random forest regression, respectively, after training on the Pareto front. As a result of using the Pareto front of the dataset for training and testing, the data variance is reduced and the regression will

[1]Datasets were last scraped on 13 September 2023.

TABLE V
ML MODEL STRUCTURE AND TRAINING RESULTS ON THE TRANSISTOR, INDUCTOR, AND CAPACITOR DATASETS GIVEN INPUTS $\bar{O}, \bar{T}$, AND OUTPUTS $\bar{X}$

Component	Inputs to ML model		Outputs $\bar{X}$	MAPE [%]
	$\bar{O}$	$\bar{T}$		
Q_i (distributor site)	$\{R_{\text{on},i}\}$	$\{V_{\text{dss},i}$, N-channel, FET technology$\}$	$\{Q_{g,i}, U_i, A_i\}$	{43, 27, 347}
Q_i (datasheet)	$\{R_{\text{on},i}\}$	$\{V_{\text{dss},i}, U_i, A_i$, N-channel, FET technology$\}$	$\{C_{\text{oss},0.1,i}, V_{\text{ds,meas},i}, \tau_{c,i}, \tau_{rr,i}\}$	{34, 17, 40, 58}
L_i	$\{f_{\text{sw}}, \Delta i_L\}$	$\{I_{\text{rated}}, I_{\text{saturation}}, L\}$	$\{R_{\text{dc}}, U, A, f_b, b, K_{fe}, \alpha, \beta, N_{\text{turns}}, A_c, V, V_{\text{core}}\}$	{27, 47, 24, 71, 10, 36, 3, 0.6, 26, 23, 33, 26}
C_i	$\{A_{Ci}\}$	$\{V_{\text{rated},i}, C_i\}$	$\{U_i\}$	{84}

Note that the transistor models are split into outputs $\bar{X}$ gathered from distributor site parameters and from PDF datasheet parameters. The models are trained using the kernel ridge regression algorithm. MAPE compares the model predictions to the corresponding parameter values for components in the testing dataset.

predict values closer to the best envelope. Fig. 9(a) shows how the inductor volume versus energy rating can be represented by a linear log–log relationship, whereas Fig. 9(b) illustrates how a more complex regression can be used to capture characteristics of available components.

In general, as shown in (2), the known inputs to the developed ML models are a combination of component specifications $\bar{T}$ based on the converter design and variables $\bar{O}$ over which the multivariate optimization algorithm iterates. The ML models need to predict the quantities $\bar{X}$ used to evaluate power loss models (such as the loss model discussed in Appendix), as well as the BOM cost U and footprint area A contributions of the considered converter transistors, inductors, and capacitors. The input/output structure of the developed ML models is summarized in Table V.

B. Selected ML Models for the Optimization Tool

Various factors were considered when selecting an ML regression algorithm for the tool: performance, run-time, application, and complexity. All models were implemented via the Scikit-learn Python package [46], and therefore, were similarly easy to implement. Linear regression was computationally the simplest and fastest, while random forest had slightly higher accuracy but longer run-time compared to other models. Ultimately, kernel ridge regression was selected for its high predictive accuracy, robustness against overfitting, and efficient run-time performance, which is essential for the optimization tool given the large number of predictions required over many iterations. Kernel ridge regression is a multivariate method that employs L2-norm regularization to address multicollinearity by limiting the magnitude of the coefficients in the linear equation [47]. Notably, kernel ridge regression is inherently continuous because it constructs predictions as a smooth combination of kernel functions, all of which are continuous. This enables the use of continuous optimization methods, which is a key distinguishing feature of the approach presented in this article.

The dataset was divided into 75% for training and validation and 25% for testing. The ML models were trained on the training data and evaluated on unseen test data using 5-fold cross-validation with randomization applied to shuffle the data points.

The regression hyperparameters were tuned using RandomizedSearchCV, an automatic hyperparameter tuner in the Scikit-learn library.

Table V shows the MAPE performance results on the testing data for transistors, inductors, and capacitors. The component cost models are based on low-volume, single-component distributor quotes. The testing results show that some values are well-predicted, while others show significant difference between the predicted and true values. These differences are expected given knowledge about the dataset as discussed in Section V-C, and are accounted for by incorporating the component selection process addressed in Section VI.

C. Model Limitations

As can be observed from the MAPE performance metrics shown in Table V, as well as the examples of regressions overlaid on the datapoints in Fig. 9, the presented ML models are subject to multiple limitations.

First, given a limited set of known input quantities $\bar{T}, \bar{O}$ used to predict any of the values in $\bar{X}$, there is inherently a high standard deviation in the output. This is reflected in the MAPE scores when evaluated on the training data itself, which were similar to the MAPE scores when evaluated on the testing data shown in Table V. As an illustration of why this phenomenon is to be expected, consider an example of N-channel Si MOSFETs. Fig. 9(b) shows an example where V_{dss} is the only known input, and it can be seen that the standard deviation in output $R_{\text{on}} * U$ is high for some V_{dss} values. To expand this concept, given an initial value of the optimization variable R_{on} and the known required voltage rating V_{dss}, one can expect that there will be significant variation in the predicted quantity, such as gate charge Q_g. All three quantities can then be used to predict the cost U, but even if Q_g is perfectly known rather than predicted, cost will vary significantly among components in the transistor dataset.

Next, one may observe a weak correlation between a transistor package (and thus, its footprint area) and its electrical parameters. This is shown in Table V, where the prediction error for the transistor footprint area is very large.

Another limitation lies in the dynamic nature of component prices and availability. This can be mitigated, in part, by periodically updating the datasets and retraining the ML models.

A key limitation lies in the quality and quantity of component data available for training. Datasheets, traditionally designed for manual use by design engineers, contain extensive but often insufficient and nonmachine-readable information. This presents

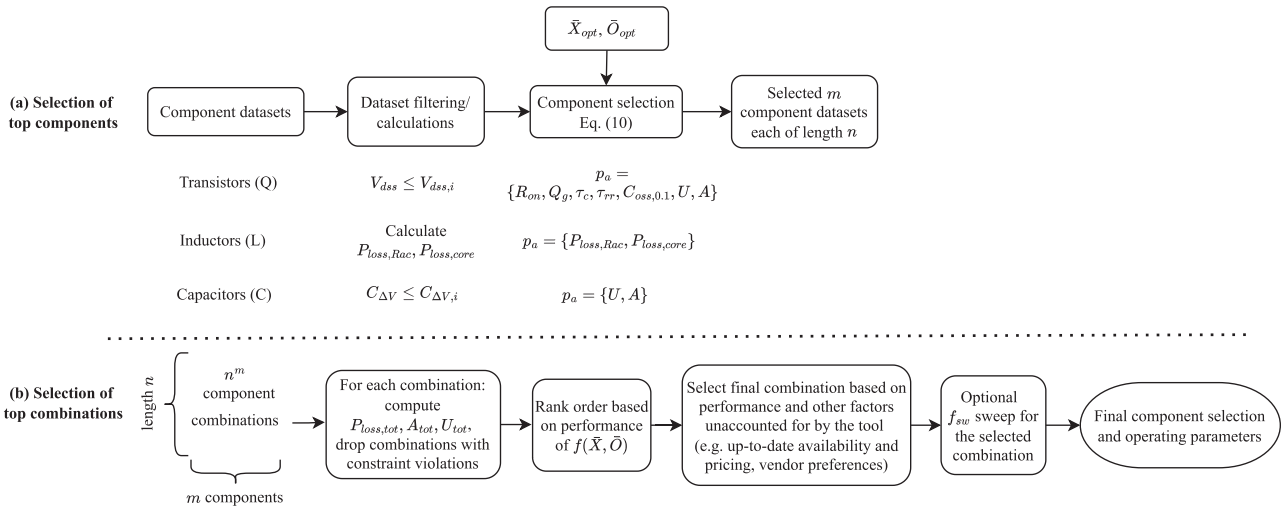

Fig. 10. Flowchart showing the process of component selection. (a) Process of selecting the top n components according to (10). (b) Exhaustive search process leading to the final component combination selection. The tool can also be used to identify the optimal switching frequency given the selected components.

substantial challenges, as discussed in Sections III and IV. Advances by manufacturers, distributors, and computer-aided design (CAD)-tool vendors, in combination with open-source initiatives such as [4], [11], are expected to improve the availability and quality of component data, enabling further developments of data-driven design approaches like the one presented in this work. Higher quality, larger datasets would enhance the prediction accuracy of all parameters, particularly those derived from scraping PDF datasheets (such as reverse-recovery parameters Q_{rr} and t_{rr}). In addition, larger datasets could make neural networks a more viable option with potential performance advantages.

VI. Component Selection

Once the optimization algorithm has converged, the optimized set of component parameters is used to create subsets of the component datasets and then perform an exhaustive search to identify a set of actual components that minimizes the objective function, as shown in Fig. 3. As previously noted, this component selection step mitigates errors associated with ML model predictions. This section describes the process by which the optimized parameter sets are used to filter the datasets to a subset of components best suited for the design, and how these subsets are used in a matrix-based exhaustive search for component selection.

A flowchart of the component selection step is shown in Fig. 10. Fig. 10(a) outlines the process for selecting the n components to include in the datasets for the exhaustive search. First, preliminary dataset filtering is performed based on requirements such as the voltage rating V_{dss} for power transistors or the capacitance $C_{\Delta V}$ required to meet the ripple requirement. Then, the components are ranked based on

$$\underset{m}{\forall} : \min_n \left[\underset{j \in l_D}{\forall} : s_j = \sum_{i=1}^{\text{len}(p_a)} \frac{(p_{a,i} - p_{a,i,\text{opt}})}{p_{a,i,\text{max}}} \right]. \tag{10}$$

Here, m is the number of components, $\min_n[S]$ is defined as a function that returns the n smallest elements from the set S, l_D is the length of each component dataset, s_j is the score for jth component in the set, p_a is the set of component parameters as shown in Fig. 10, and $p_{a,i,\text{opt}}$ is taken from the set of optimized parameters $\bar{O}_{\text{opt}}, \bar{X}_{\text{opt}}$. One may note that taking an absolute value $|p_a - p_{a,\text{opt}}|$ in (10) might result in selecting a component with parameters closer to the optimized set. However, since a preferred component should have parameters at least as good as the optimized parameters, the absolute value is not taken in (10). The number n of components to retain presents a tradeoff between the execution time for the exhaustive search, and how close the end result may be to the minimum of the considered objective function. This tradeoff is examined in the design examples in the next section. Fig. 10(b) depicts the search process conducted across the n^m possible component combinations to identify the highest-performing designs that satisfy the design constraints. Subsequently, a sweep over the switching frequency can be performed to determine the optimal frequency for the selected component combination.

VII. Design Examples

The tool described in Sections II–VI is used to design 48 V-to-12 V, 5 A synchronous buck converters using Si or GaN transistors, based on Fig. 2 and according to the specifications and constraints shown in Table VI. The optimization problem is defined in (1), with the objective function defined as the total power loss

$$f(\bar{X}, \bar{O}) = P_{\text{loss,total}} \tag{11}$$

where the power loss model is described in Appendix, Fig. 13 shows the flowchart in Fig. 3 filled in with the inputs and results analysis of this design example. The lists of component indices are

$$\bar{Q} = [Q_1, Q_2],\ \bar{L} = [L_1],\ \bar{C} = [C_1, C_2]. \tag{12}$$

TABLE VI
SYNCHRONOUS BUCK CONVERTER EXAMPLE SPECIFICATIONS AND CONSTRAINTS

Constraint	Value
V_{in}	48 V
$V_{\text{dss,required}}$	60 V
V_{out}	12 V
I_{out}	5 A
V_{GG}	10 V (Si) or 5 V (GaN)
$U_{\text{max,total}}$	\$5–\$11
$A_{\text{max,total}}$	800 mm^2
$\Delta i_L/I_L$	70%

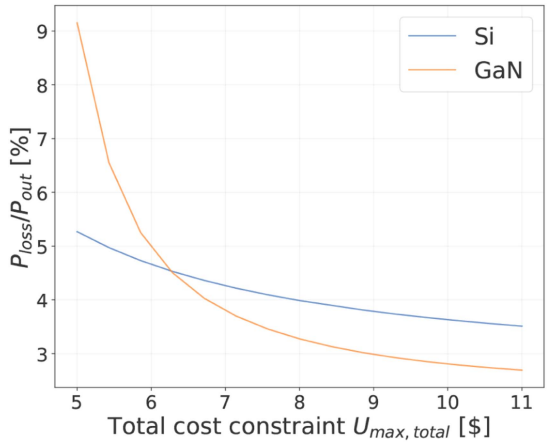

Fig. 11. Results of the optimization step of the design tool given a range of $U_{max,total}$ for the buck converter example designed using Si or GaN power FETs. P_{loss}/P_{out} is $f(\bar{X},\bar{O})$ evaluated at $\bar{X}_{opt},\bar{O}_{opt}$.

The converter specifications and constraints are

$$\bar{T}_{\text{con}} = [V_{\text{in}},\ V_{\text{out}},\ I_{\text{out}},\ V_{\text{dss}},\ V_{\text{GG}}\ U_{\text{max,total}},\ A_{\text{max,total}},\ (\Delta i_L/I_L)_{\text{max}}]. \quad (13)$$

The transistor, inductor, and capacitor inputs $\bar{T}$, optimization variables $\bar{O}$, and outputs $\bar{X}$ are shown in Table V, where U is the BOM cost and A is the BOM footprint area. At this stage in the tool's development, all other converter components (gate drivers, controller) are assumed to collectively have approximately the same impact on the converter losses, size, and cost in all considered designs, and are therefore excluded from calculations of $f(\bar{X},\bar{O})$ and the constraints.

This section is organized as follows. Section VII-A runs the design tool for the specified buck converter example and selects four specific design points to complete the component selection. Section VII-B discusses the experimental testing results on the prototypes built using the selected components.

A. Design Example Implementation

To examine the impact of the BOM cost constraint $U_{\text{max,total}}$ on the achievable power loss, the design tool is run repeatedly over a range of $U_{\text{max,total}}$ from \$5 to \$11, and for two transistor types, silicon (Si) and GaN. The BOM footprint area constraint is kept constant, as shown in Table VI. On a standard desktop, the algorithm converges in less than one minute for each run. Fig. 11 shows the power loss predicted based on the optimized component parameters as a function of the increasing cost constraint $U_{\text{max,total}}$. As expected, the optimized power loss decreases as the BOM cost constraint is increased, until the losses reach a lower limit. Also, a minimum cost constraint is necessary for the tool to converge to a feasible design. A comparison is made between Si- and GaN-based optimized designs. One may note how the GaN-based designs outperform Si-based designs as the cost constraint is increased. Fig. 11 also demonstrates the tool's multiobjective optimization capabilities by repeatedly running it with varying constraints. The figure presents a Pareto front illustrating the tradeoff between power loss and cost for two different transistor technologies. The results show that Si-based designs achieve lower losses under strict cost constraints, whereas GaN transistors offer higher efficiency as cost constraints are eased.

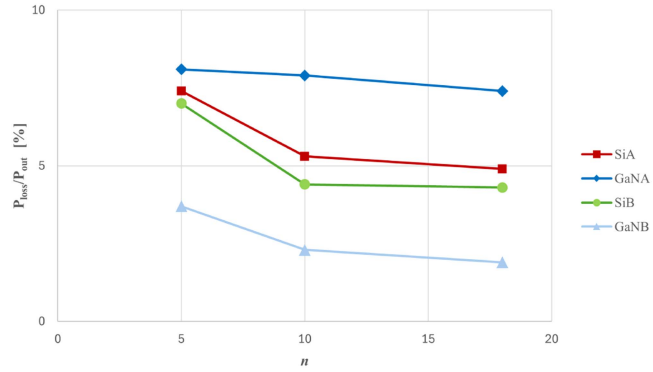

Fig. 12. Predicted P_{loss}/P_{out} as a function of the number n of highestscoring components per dataset selected to perform the exhaustive search step. The P_{loss}/P_{out} associated with the component combination leading to the lowest loss is shown here at the switching frequency fsw returned by the optimization algorithm.

TABLE VII
$\bar{O}_{\text{OPT}}$: RESULTS FOR $\bar{O}$ AFTER THE OPTIMIZATION ALGORITHM CONVERGED

	Si		GaN	
	A	B	A	B
$P_{\text{loss}}/P_{\text{out}}$ [%]	5.3	3.5	9.0	2.7
$R_{\text{on},Q1}$ [mΩ]	10	6	20	7
$R_{\text{on},Q2}$ [mΩ]	13	6	48	4
f_{sw} [kHz]	150	50	1500	300
$\Delta i_L/I_L$	0.37	0.29	0.26	0.09
A_{C1} [mm^2]	11	9	9	10
A_{C2} [mm^2]	8	10	11	7

Next, two specific cost constraints are used to examine the Si- and GaN-based designs: \$5 (Si$_A$ and GaN$_A$ designs) and \$11 (Si$_B$ and GaN$_B$ designs). Table VII shows the converged values of the optimization variables. One may observe that the trends in the optimization variables match expectations, because more expensive designs allow for lower switching frequencies, reduced transistor on-resistances, and smaller inductor current ripple. One may note the differences in parameter breakdown between different technologies under varying cost constraints.

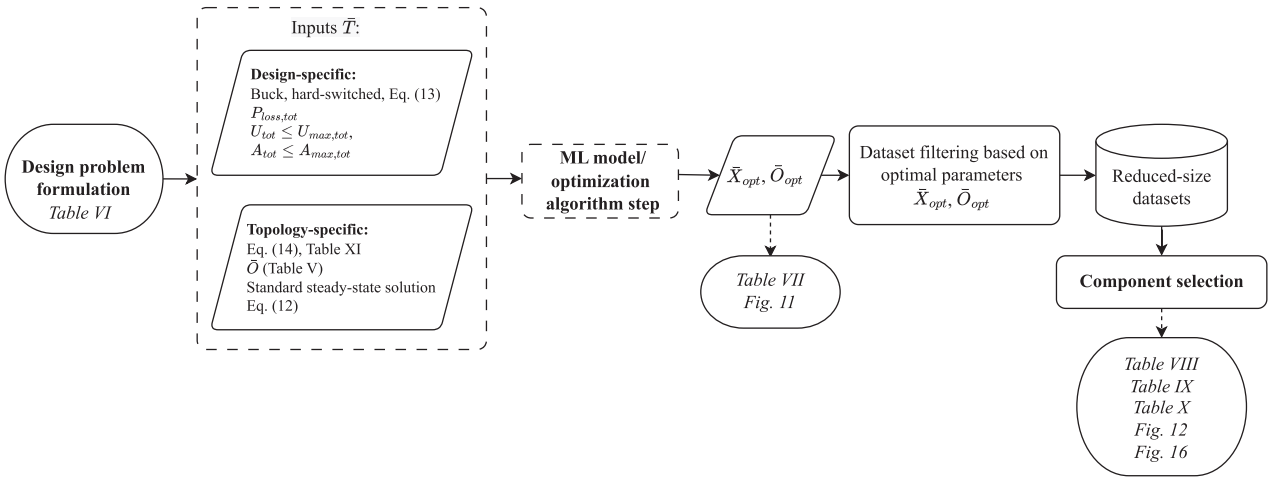

Fig. 13. Design approach flowchart, similar to that of Fig. 3, now with the specifications and the results for the presented design examples.

One counterintuitive result is that the optimization converges to $R_{\text{on},1} < R_{\text{on},2}$ for Si and GaN designs with the smaller cost constraint. One would expect the opposite given that the converter operates at relatively low duty cycle $D \approx 12/48 = 0.25$, and the Q_2 rms current is larger than the Q_1 rms current. A closer inspection shows that the designs with reduced cost constraints are optimized with $R_{\text{on},1} < R_{\text{on},2}$ because a lower budget is available for inductors, which, in turn, leads to higher optimized switching frequency under the more restrictive cost constraint. At higher switching frequencies, Q_1 transistor turn-OFF loss P_{Qoff} becomes more significant, as discussed in Appendix. The increase in this loss mechanism tends to favor a larger Q_1 device with a smaller $R_{\text{on},1}$ and larger C_{oss}, leading to an overall lower total power loss.

The four specific design points Si_A, Si_B, GaN_A, GaN_B, were used to complete the design approach. For each design, the optimized parameters $\bar{X}, \bar{O}$ were used to filter the datasets and perform the component selection, as described in Section VI. Fig. 12 shows the predicted power loss performance for various values of n components retained in the reduced datasets for transistors, inductors, and capacitors. The component selection step of the tool was chosen to consider $n = 10$ components, determined by their s_j scores that yielded the best performance. This results in a search run-time of 2–4 min on a standard desktop. This n value represents a good tradeoff between run-time and the selected components leading to a highly efficient converter. Increasing the run-time slightly (from < 1 min. to 2–4 min.) results in a significant improvement in predicted converter efficiency as the tool is now able to see a sufficiently large number of component combinations even in the context of the specific loss equations, whereas a significant increase in run-time (from 2–4 to 20–30 min.) leads to a relatively small predicted efficiency improvement. These results show how the ML model-based component parameter recommendations are a good starting point for component selection, and that the automatic component selection step can help mitigate errors in the parameter recommendations that occur as a result of ML model errors and limitations.

The final step is the f_{sw} sweep given the selected components. The results for each of the four designs are summarized in Table VIII, where one may observe how adjusting f_{sw} yields small further improvements in power loss.

TABLE VIII
SWITCHING FREQUENCIES AND CORRELATED POWER LOSS AT THE CONVERGED f_{sw} VERSUS AFTER THE f_{sw} SWEEP

Design	Step	f_{sw} [kHz]	$P_{\text{loss}}/P_{\text{out}}$ [%]
Si_A	Algorithm	150	5.3
	Sweep	100	4.9
GaN_A	Algorithm	1500	7.9
	Sweep	2500	7.0
Si_B	Algorithm	50	4.4
	Sweep	100	4.3
GaN_B	Algorithm	330	2.3
	Sweep	200	2.2

In both cases, the $P_{\text{loss}}/P_{\text{out}}$ calculations are made given the parameters of the selected components.

Table IX shows the break-down of individual loss, cost, and area contributions of Q_1, Q_2, L_1, C_1, and C_2 at the selected design points. The table compares the loss breakdown of the optimized parameters determined by the ML-based optimization step against the loss breakdown given the datasheet parameters of the selected components. It can be seen that the breakdowns are similar between the two results. In most cases, the largest contributor to overall loss, cost, and area is the inductor.

B. Hardware Prototypes

The selected components listed in Table X were used to build experimental hardware prototypes. The power stages of the assembled boards are shown in Fig. 14, and typical operating waveforms are shown in Fig. 15. Each prototype was tested across a range of operating frequency f_{sw} and output power P_{out}. Fig. 16 compares the experimentally measured total power loss against the predicted total power loss, calculated using the selected components and the optimized operating conditions. The predicted loss is broken down into individual loss contributions. One may note how the experimentally measured losses qualitatively match the predicted trends between the four

TABLE IX
LOSS, COST, AND AREA CONTRIBUTION BREAKDOWN BY COMPONENT AND SPECIFIC LOSS CONTRIBUTION

			Si				GaN			
			A ($\leq$ \$5)		B ($\leq$ \$11)		A ($\leq$ \$5)		B ($\leq$ \$11)	
Attribute		Component	X_{opt}	Selected	X_{opt}	Selected	X_{opt}	Selected	X_{opt}	Selected
Loss [W]	P_{Qc}	Q_1	0.10	0.11	0.05	0.11	0.18	0.16	0.04	0.06
		Q_2	0.41	0.34	0.18	0.36	1.35	0.47	0.18	0.17
	P_{Qq}	Q_1	0.03	0.03	0.01	0.009	0.03	0.07	0.03	0.03
		Q_2	0.02	0.03	0.01	0.009	0.01	0.07	0.02	0.03
	P_{Qoff}	Q_1	0.12	0.05	0.02	0.05	1.99	0.70	0.08	0.05
	P_{Qrr}	Q_1	0.57	0.76	0.27	0.17	0	0	0	0
	P_{Qds}	Q_1	0.02	0.05	0.02	0.02	0.07	0.31	0.10	0.11
	$P_{\text{Lw,dc}}$	L_1	0.55	1.2	0.74	0.52	0.36	0.14	0.62	0.18
	$P_{\text{Lw,ac}}$	L_1	0.38	0.12	0.11	0.20	0.17	0.27	0.04	0.13
	P_{Lc}	L_1	0.70	0.21	0.45	0.27	1.00	2.24	0.27	0.41
	P_{PCB}		0.21	0.25	0.25	0.27	0.26	0.31	0.25	0.26
		Total	3.12	3.15	2.11	2.63	5.42	4.73	1.63	1.40
Cost [\$]		Q_1	0.71	0.57	1.13	0.94	1.89	1.58	3.14	2.53
		Q_2	0.60	0.57	0.99	0.57	1.61	1.58	2.64	2.53
		L_1	3.11	1.97	8.16	7.37	1.10	0.41	4.76	4.02
		C_1	0.30	0.66	0.38	0.91	0.22	0.66	0.26	0.91
		C_2	0.28	0.26	0.34	0.96	0.18	0.26	0.19	0.96
		Total	5.00	4.03	11.00	10.75	5.00	4.49	11.00	10.95
Area [mm^2]		Q_1	16	11	19	11	0.81	0.81	0.81	0.81
		Q_2	18	11	20	11	0.81	0.81	0.81	0.81
		L_1	197	301	511	517	52	44	307	256
		C_1	11	5	9	8	9	5	10	8
		C_2	8	3	10	5	11	3	9	5
		Total	250	331	569	552	74	54	328	270

The expected performance was generated for the synchronous buck converter examples with the cost constraint $U_{\text{max,total}}$ equal to \$5 or \$11, using Si or GaN MOSFETs, given the commercially available components shown in Table X. In a few instances, due to availability, a component with attributes similar to the recommended component was selected for the prototype. All calculations shown here are completed using f_{sw} returned by the optimization step. One may note the similarities between the results from the optimization step and those with the selected components. Appendix A discusses each power loss contribution for the buck converter example in detail.

TABLE X
SELECTED COMPONENTS FOR THE PROTOTYPES SHOWN IN FIG. 14

Design	Component	Mfr. part no.	Parameters
Si_A	Q_1	PXN012-60QLJ	V_{dss} = 60V, R_{on} = 11.5 mΩ
	Q_2	PXN012-60QLJ	V_{dss} = 60V, R_{on} = 11.5 mΩ
	L_1	TMPA 1707SP-330MN-D	L = 33 μH
	C_1	GRM31CC72A475ME11L	C = 4.7 μF
	C_2	CL21A226MAYNNNE	C = 22 μF
GaN_A	Q_1	EPC2052	V_{dss} = 80V, R_{on} = 13.5 mΩ
	Q_2	EPC2052	V_{dss} = 80V, R_{on} = 13.5 mΩ
	L_1	240-MGAH0603R68M-10CT-ND	L = 0.68 μH
	C_1	GRM31CC72A475ME11L	C = 4.7 μF
	C_2	CL21A226MAYNNNE	C = 22 μF
Si_B	Q_1	SIS176LDN-T1-GE3	V_{dss} = 70V, R_{on} = 10.9 mΩ
	Q_2	PXN012-60QLJ	V_{dss} = 60V, R_{on} = 11.5 mΩ
	L_1	AMDLA2213Q-470MT	L = 47 μH
	C_1	GRM32EC72A106ME05L	C = 10 μF
	C_2	C3216X5R1E476M160AC	C = 47 μF
GaN_B	Q_1	EPC2204	V_{dss} = 100V, R_{on} = 5.6 mΩ
	Q_2	EPC2204	V_{dss} = 100V, R_{on} = 5.6 mΩ
	L_1	SRP1510CA-100M	L = 10 μH
	C_1	GRM32EC72A106ME05L	C = 10 μF
	C_2	3216X5R1E336M160AC	C = 33 μF

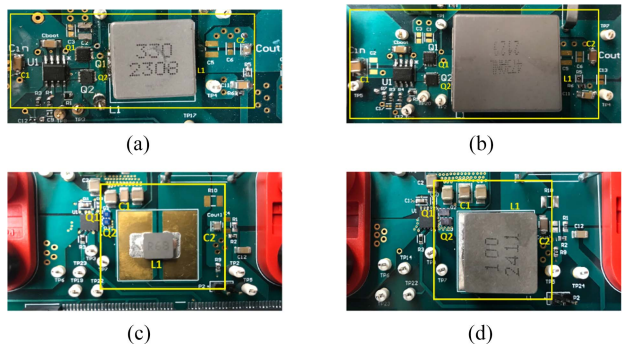

Fig. 14. Experimental prototypes. (a) Si_A. (b) Si_B. (c) GaN_A. (d) GaN_B.

designs: GaN_A had the highest loss, then Si_A, then Si_B, then GaN_B. However, the measured losses are somewhat higher than the predicted losses in Fig. 16 due to inaccuracies in the presented loss characterization for the buck converter.

The difference in efficiencies between GaN_B and Si_B was not as high as expected. Since more of the cost budget was spent on GaN components, less was available to be spent on the inductor. After analyzing the results, it was determined that most of the discrepancy in the loss model is due to terms involving the switching frequency f_{sw}, i.e., inductor ac losses and switching losses. In particular, the inductor temperature rise,

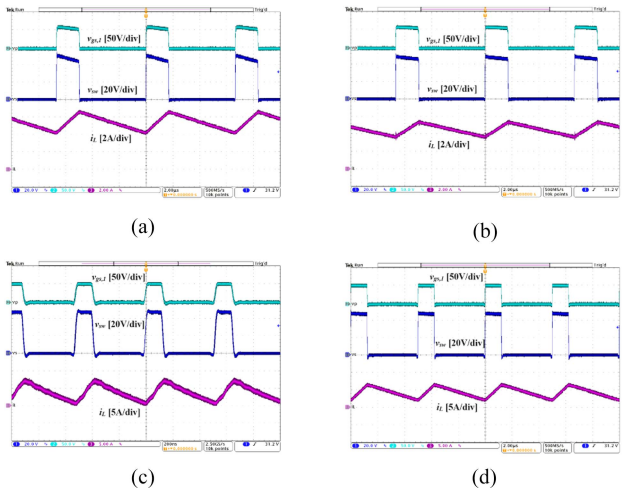

Fig. 15. Operating waveforms. (a) Si_A. (b) Si_B. (c) GaN_A. (d) GaN_B.

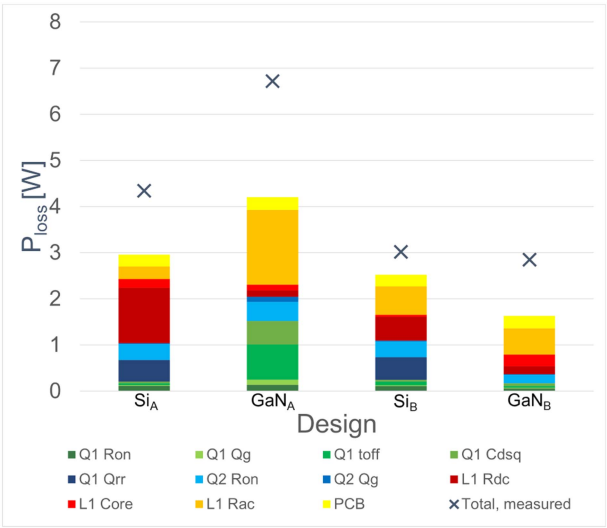

Fig. 16. Comparison between experimentally measured losses and losses predicted by the loss model described in Appendix for each of the four prototypes: Si_A, GaN_A, Si_B, GaN_B.

especially in cases of high $\Delta i_L/I_L$, suggests that the inductor ac winding loss estimation method could be improved. For example, methods developed in [6] could improve the inductor core loss estimation. Other differences in the observed efficiency versus the predicted efficiency can be attributed to Q_1 losses, particularly the estimation of $C_{\mathrm{ds},q}$, which uses an estimated $C_{\mathrm{oss},0.1}$ that differs from the datasheet-based value, and the Q_{rr} loss estimation, which uses estimated τ_c and τ_{rr} parameters. It is noted that the evaluation of design-oriented parameters and the power loss equation discussed in Section IV and Appendix are both simplified estimation approaches, which contribute to the differences between theoretical and measured losses.

Overall, from the results shown in Fig. 16, it can be concluded that the ML-based converter design method developed in this work can be useful in rapidly determining high-performing designs. It is noted that additional measures can be taken to improve the theoretical loss models used in the tool. Additional design examples, illustrating the generalized nature of the tool, can be found in [21].

VIII. Conclusion

This work presents a data-driven approach, implemented as a tool, to assist designers in quickly selecting components for converter designs that achieve high performance across multiple metrics, including efficiency, cost, and size. The approach leverages a unique combination of data acquisition and ML-based regression models for components, enabling efficient optimization over a large design space through a continuous optimization process.

Component data is gathered from distributor websites on 22 000 power transistors, 48 000 fixed inductors, and 110 000 Class II MLC capacitors. In addition, automated scraping of PDF datasheets is used to augment the power FET dataset with additional datasheet parameters, including device capacitance and body-diode reverse recovery parameters. Methods are also introduced for component data preprocessing, including outlier removal and Pareto dominance, to improve component data quality and relevance for the power converter design. Novel automated methods combining data-driven and physics-based approaches are introduced to enhance component datasets with design-oriented parameters essential for power loss modeling. For power FETs, these parameters include the temperature dependence of transistor on-resistance, equivalent charge capacitance, and body-diode reverse recovery parameters across various operating conditions. For fixed inductors, predictive models are developed to account for core losses and ac copper losses, while the dependence of capacitance on dc bias is modeled for Class II MLC capacitors.

Trends in component parameters are effectively captured using simple regression-based ML models that are relatively easy to train on moderatly sized datasets and require minimal computational resources to make predictions. In particular, kernel ridge regression is selected, which offers high predictive accuracy, robustness against overfitting, continuous predictions, and efficient run-time performance. The continuity and efficiency are important, as these ML-based component models are frequently used during the optimization process. Furthermore, the one-time training of these models allows them to be reused universally across the design of various converter topologies. The developed ML models support continuous optimization over a large design space, enabling relatively fast and robust convergence. The results of the optimization step provide valuable insights into the influence of component parameters while reducing sensitivity to specific component availability. Finally, the optimized component parameters are used to filter the large component datasets to allow for a computationally efficient search leading to the component selection. The results are sets of commercially available part numbers rank ordered by performance, assisting designers in meeting their design objectives efficiently.

The developed tool is applied to the design of 48-to-12 V, 5 A synchronous buck dc–dc converters using either Si or GaN transistors, where the objective is to minimize loss while meeting BOM cost and area constraints. As an example of

the tool's multiobjective optimization capabilities, Pareto fronts are generated that illustrate the tradeoffs between achievable loss and cost. The results reveal that Si-based designs tend to exhibit lower losses when cost constraints are tight, while GaN transistors deliver superior efficiency as the cost constraint is relaxed. Four designs—GaN- and Si-based under both low and high cost constraints—are selected to demonstrate how the tool recommends specific sets of component part numbers. These designs are then implemented as experimental prototypes, validating the efficiency, cost, and size trends predicted by the tool.

The complete code for the tool, encompassing data acquisition and processing, ML modeling, optimization, and component selection steps, is published to support further open-source development [23]. Limitations of the approach and the developed tool are highlighted, especially pointing to the need for more comprehensive, machine-readable component data, which would enable further developments of data-driven design automation tools for power electronics.

Appendix
Loss Model for a Synchronous Buck Converter

This section describes a loss model for the buck converter shown in Fig. 2. The converter operates from V_{in} input dc voltage and produces V_{out} dc output voltage at I_{out} dc output current. It is assumed that the converter operates in continuous conduction mode with inductor current ripple $\Delta i < I_{\text{out}}$.

As shown in Table XI, the conduction loss P_{Qc} of the MOSFETs Q_1 and Q_2 is obtained using the standard expression, with the multiplier k_T to account for the anticipated temperature rise. Estimation of k_T is discussed in Section IV-A.

Estimation of the switching loss P_{Qrr} due to the reverse recovery of the body diode of the synchronous rectifier Q_2 is challenging. Power MOSFET datasheets most often include a single value $Q_{\text{rr},d}$ for the reverse recovery charge, obtained for recovery from conducting current $I_{F,d}$ at a certain current decay rate $S_d = (di/dt)_d$, as well as single values for the reverse recovery time $t_{\text{rr},d}$, and the amplitude $I_{\text{rr},d}$ of reverse current during recovery. Starting from the datasheet-reported values $Q_{\text{rr},d}$, $I_{F,d}$, S_d, $t_{\text{rr},d}$, $I_{\text{rr},d}$, the diode reverse recovery model [40] can be used to determine the time constants associated with the pn-junction: effective carrier lifetime τ_c and time constant τ_{rr} of the current decay when the junction is reverse-biased. Once $\tau_{c,d}$ and $\tau_{\text{rr},d}$ are determined, the model can be used to calculate Q_{rr}, t_{rr}, and I_{rr} at different operating conditions characterized by I_F and S. Details of this datasheet-based reverse-recovery model can be found in [34] and [40].

In this work, the datasheet-reported values $Q_{\text{rr},d}$, $t_{\text{rr},d}$, S_d, and $I_{F,d}$ have been scraped from all power MOSFETs in the datasheet-based dataset. The corresponding calculated values of $\tau_{c,d}$ and $\tau_{\text{rr},d}$ are used to train ML models for power MOSFETs, as discussed in Section V and summarized in Table V. In the optimization step, the ML model-predicted values for τ_c and τ_{rr} are used to calculate Q_{rr}, t_{rr} and then the loss contribution P_{Qrr} as shown in Table XI, based on the operating conditions in the considered converter design. In the buck converter example, $I_F = I_{\text{out}} - \Delta i$, while the operating decay rate S has an assumed value of 100 $\text{A}/\mu\text{s}$. One may note that the expression for P_{Qrr} in Table XI assumes a "snappy" diode, which tends to overestimate the reverse-recovery related loss on Q_1.

TABLE XI
Loss Model Equations for the Buck Converter Based on Datasheet Parameters and the Design-Oriented Parameters Estimated as Described in Section IV

Loss symbol	Loss description	Modeling equation
P_{Qc}	Conduction	$\sum_{i=1,2} k_T R_{\text{on},i} I^2_{\text{rms},i}$
P_{Qrr}	Reverse-recovery charge	$(I_F t_{\text{rr}} + Q_{\text{rr}})\, V_{\text{in}}\, f_{\text{sw}}$
P_{Qds}	Charge-equivalent capacitance	$C_{\text{ds},Q} V^2_{\text{in}}\, f_{\text{sw}}$
P_{Qoff}	Off-transition	$\frac{I^2_{\text{off},Q1} t^2_{\text{off}}}{48 C_{\text{oss},0.1}} f_{\text{sw}}$
P_{Qg}	Gate-charge	$\sum_{i=1,2} Q_{g,i} V_{\text{GG}}\, f_{\text{sw}}$
P_{Lw}	Winding	$R_{\text{dc}} I^2_{\text{out}} + \sum_{k=1}^{11} R_{\text{ac}}(k f_{\text{sw}}) I^2_{\text{rms},k}$
P_{Lc}	Core	$V_c \cdot \text{iGSE}(\mathbf{B}, \mathbf{t}, K_{fe}, \alpha, \beta)$
P_{PCB}	PCB	$R_{\text{PCB}}\, I^2_{L,\text{rms}}\, r_T$

Hard discharging of Q_1 output capacitance and charging of Q_2 output capacitance contributes to the switching loss P_{Qds}, which can be modeled using the charge-equivalent capacitance $C_{\text{ds},Q}$ [37], [38], [41]. The automated estimation of $C_{\text{ds},Q}$ across the MOSFET dataset is discussed in Section IV-C. Note that one assumption made in this estimation method is that $C_{\text{oss},0.1}$ is assumed to be equivalent between Q_1 and Q_2, and $C_{\text{oss},0.1}$ of Q_1 is used in calculations.

An overlap in device voltage and current during the Q_1 switch-OFF transition leads to a P_{Qoff} switching loss, which assumes a linear decay of current in the Q_1 channel from $I_{\text{off},Q1} = I_{\text{out}} + \Delta i$ to zero over the interval t_{off}. This results in an increase in voltage across the output capacitance of the device, approximated as $C_{\text{oss},0.1}$. Here, t_{off} can be considered a constant for a given device voltage rating and type, and $C_{\text{oss},0.1}$ is the device output capacitance C_{oss} at V_{ds} equal to 10% of the rated voltage. Automated estimation of $C_{\text{oss},0.1}$ across the MOSFET dataset is discussed in Section IV-C.

The inductor loss model reported in Table XI includes the inductor winding loss P_{Lw} due to the dc resistance R_{dc} reported in datasheets, as well as the loss due to the winding ac resistance as a function of frequency $R_{\text{ac}}(f)$ in (8) ($m = 11$ current harmonics are considered in the loss model calculations). Furthermore, inductor core loss P_{Lc} is estimated using the improved generalized Steinmetz equation (iGSE) approach [48]. For the inductor in the buck example, the inputs are the flux density waveform $\mathbf{B} = [-\Delta B, +\Delta B, -\Delta B]$, $\mathbf{t} = [0, \text{DT}_s, T_s]$, $\Delta B = L\Delta i_L/(N_{\text{turns}} A_c)$, and core volume V_c. Automated estimation of the design-oriented parameters needed for the inductor loss model is discussed in Section IV-E.

Finally, the total converter loss is obtained as

$$P_{\text{loss,total}} = P_{Qc} + P_{\text{Qrr}} + P_{\text{Qds}} + P_{\text{Qoff}} + P_{Qg} + P_{\text{Lw}} + P_{\text{Lc}} + P_{\text{PCB}}. \quad (14)$$

ACKNOWLEDGMENT

Any opinion, findings, and conclusions or recommendations expressed in this article are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] F. Lin, X. Li, X. Zhang, and H. Ma, "Star: One-stop optimization for dual-active-bridge converter with robustness to operational diversity," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 12, no. 3, pp. 2758–2773, Jun. 2024.

[2] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Trans. Power Electron.*, vol. 36, no. 4, pp. 4633–4658, Apr. 2021.

[3] T. Guillod, P. Papamanolis, and J. W. Kolar, "Artificial neural network (ANN) based fast and accurate inductor modeling and design," *IEEE Open J. Power Electron.*, vol. 1, pp. 284–299, 2020.

[4] H. Li et al., "MagNet: An open-source database for data-driven magnetic core loss modeling," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2022, pp. 588–595.

[5] H. Li et al., "How MagNet: Machine learning framework for modeling power magnetic material characteristics," *IEEE Trans. Power Electron.*, vol. 38, no. 12, pp. 15829–15853, Dec. 2023.

[6] M. Chen et al., "MagNet challenge for data-driven power magnetics modeling," *IEEE Open J. Power Electron.*, to be published, doi: 10.1109/OJPEL.2024.3469916.

[7] G. Kruse, D. Happel, S. Ditze, S. Ehrlich, and A. Rosskopf, "Parameter optimization of LLC-converter with multiple operation points using reinforcement learning," in *Proc. IEEE 20th Biennial Conf. Electromagn. Field Comput.*, 2022, pp. 1–2.

[8] F. Tian, D. B. Cobaleda, H. Wouters, and W. Martinez, "Parameter design optimization for DC-DC power converters with deep reinforcement learning," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2022, pp. 1–7.

[9] X. Li, X. Zhang, F. Lin, and F. Blaabjerg, "Artificial-intelligence-based design for circuit parameters of power converters," *IEEE Trans. Ind. Electron.*, vol. 69, no. 11, pp. 11144–11155, Nov. 2022.

[10] F. Tian, D. B. Cobaleda, and W. Martinez, "Artificial-intelligence based DC-DC converter efficiency modelling and parameters optimization," in *Proc. 24th Eur. Conf. Power Electron. Appl.*, 2022, pp. 1–7.

[11] N. Förster, P. Rehlaender, O. Wallscheid, F. Schafmeister, and J. Böcker, "An open-source transistor database and toolbox as a unified software engineering tool for managing and evaluating power transistors," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2022, pp. 1747–1751.

[12] F. Tian, Q. Sui, D. B. Cobaleda, and W. Martinez, "Automated extraction of data from MOSFET datasheets for power converter design automation," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 12, no. 6, pp. 5648–5660, Dec. 2024.

[13] W. Martinez and C. Cortes, "Improved design automation combinatorial algorithms for multi-objective optimization of high voltage-gain DC-DC converters," in *Proc. 23rd Eur. Conf. Power Electron. Appl.*, 2021, pp. P.1–P.8.

[14] F. Tian, D. B. Cobaleda, and W. Martinez, "Deep reinforcement learning for DC-DC converter parameters optimization," in *Proc. IEEE 31st Int. Symp. Ind. Electron.*, 2022, pp. 325–330.

[15] S. Wang et al., "An intelligent system for automatic selection of DC-DC converter topology with optimal design," in *Proc. AAAI Workshop AI Des. Manuf.*, 2022.

[16] M. Zhang, X. Wang, and Q. Xu, "Data-driven modeling of power-electronics-based power system considering the operating point variation," in *Proc. IEEE Energy Convers. Congr. Expo.*, 2021, pp. 3513–3517.

[17] K. J. Goodrick, J. Zhu, and D. Maksimović, "Systematic optimization of multiple-output DC distribution architectures," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, vol. 9, no. 5, pp. 5703–5717, Oct. 2021.

[18] S. Reese and D. Maksimovic, "An approach to DC-DC converter optimization using machine learning-based component models," in *Proc. IEEE 23rd Workshop Control Model. Power Electron.*, 2022, pp. 1–8.

[19] S. Reese et al., "Loss estimation and design of DC-DC converters using physics- and data-based component models," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2023, pp. 82–89.

[20] K. J. Goodrick, A. Butler, T. Byrd, and D. Maksimović, "Machine learning estimators for power electronics design and optimization," in *Proc. IEEE Des. Methodol. Conf.*, 2021, pp. 1–8.

[21] S. Reese, B. Sauter, A. Kumar, S. Hu, V. Iyer, and D. Maksimovic, "Machine learning-aided design of switched-mode power converters," in *Proc. IEEE Energy Convers. Congr. Expo*, 2024, pp. 1–7.

[22] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour D'horizon," *Eur. J. Oper. Res.*, vol. 290, no. 2, pp. 405–421, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221720306895

[23] S. Reese, "CoPEC data-driven component level design," 2025. Accessed: Jan. 18, 2025. [Online]. Available: https://github.com/Colorado-Power-Electronics-Center/CoPEC-data-driven component-level-design

[24] M. J. D. Powell, "A view of algorithms for optimization without derivatives," CMS, Univ. of Cambridge, Cambridge, CB3 0WA, U.K., Tech. Rep. DAMTP 2007/NA03, Apr. 2007.

[25] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[26] "Selenium project contributors, selenium WebDriver, selenium project, open-source browser automation framework," 2024. [Online]. Available: https://www.selenium.dev/documentation/webdriver/

[27] Digi-Key Electronics, "Digi-key electronics, global distributor of electronic components and automation products," 2024. [Online]. Available: https://www.digikey.com

[28] Mouser Electronics, "Mouser Electronics, global distributor of semiconductors and electronic components," 2024. [Online]. Available: https://www.mouser.com

[29] L. Richardson, "Beautiful soup: A python library for web scraping, crummy, LLC, python library for parsing HTML and XML, often used for web scraping tasks," 2024. [Online]. Available: https://www.crummy.com/software/BeautifulSoup/

[30] G. Mclachlan, "Mahalanobis distance," *Resonance*, vol. 4, pp. 20–26, Jun. 1999.

[31] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[32] Y. Xu, C. N. M. Ho, A. Ghosh, and D. Muthumuni, "A datasheet-based behavioral model of SiC MOSFET for power loss prediction in electromagnetic transient simulation," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2019, pp. 521–526.

[33] F. Wang, S. Kher, T. Fichtner, and J. Aurich, "A new power MOSFET model and an easy to use characterization tool using device datasheet," in *Proc. IEEE 14th Workshop Control Model. Power Electron.*, 2013, pp. 1–5.

[34] D. Christen and J. Biela, "Analytical switching loss modeling based on datasheet parameters for MOSFETs in a half-bridge," *IEEE Trans. Power Electron.*, vol. 34, no. 4, pp. 3700–3710, Apr. 2019.

[35] A. Andreta, Y. Lembeye, L. L. Villa, and J.-C. Crebier, "Statistical modelling method for active power components based on datasheet information," in *Proc. Int. Exhib. Conf. Power Electron., Intell. Motion, Renewable Energy Manage.*, 2018, pp. 1–7.

[36] M. Mudholkar, S. Ahmed, M. N. Ericson, S. S. Frank, C. L. Britton, and H. A. Mantooth, "Datasheet driven silicon carbide power MOSFET model," *IEEE Trans. Power Electron.*, vol. 29, no. 5, pp. 2220–2228, May 2014.

[37] M. Kasper, R. M. Burkart, G. Deboy, and J. W. Kolar, "ZVS of power MOSFETs revisited," *IEEE Trans. Power Electron.*, vol. 31, no. 12, pp. 8063–8067, Dec. 2016.

[38] J. Azurza, G. Zulauf, J. Kolar, and G. Deboy, "New figure-of-merit combining semiconductor and multi-level converter properties," *IEEE Open J. Power Electron.*, vol. 1, pp. 322–338, 2020.

[39] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231216003325

[40] P. Lauritzen and C. Ma, "A simple diode model with reverse recovery," *IEEE Trans. Power Electron.*, vol. 6, no. 2, pp. 188–191, Apr. 1991.

[41] D. Costinett, D. Maksimovic, and R. Zane, "Circuit-oriented treatment of nonlinear capacitances in switched-mode power supplies," *IEEE Trans. Power Electron.*, vol. 30, no. 2, pp. 985–995, Feb. 2015.

[42] R. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*, 3rd ed. Berlin, Germany: Springer, 2020.

[43] C. R. Sullivan, B. A. Reese, A. L. F. Stein, and P. A. Kyaw, "On size and magnetics: Why small efficient power inductors are rare," in *Proc. Int. Symp. 3D Power Electron. Integration Manuf.*, 2016, pp. 1–23.

[44] S. Reese, T. Byrd, J. Haddon, and D. Maksimovic, "Machine learning-based component figures of merit and models for DC-DC converter design," in *Proc. IEEE Des. Methodol. Conf.*, 2022, pp. 1–6.

[45] Magnetics, "Powder cores," 2023. [Online]. Available: https://www.mag-inc.com/Products/Powder-Cores
[46] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[47] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012, ch. 14.4.3, pp. 492–493. [Online]. Available: https://scikit-learn.org/stable/modules/kernel_ridge.html
[48] K. Venkatachalam, C. Sullivan, T. Abdallah, and H. Tacca, "Accurate prediction of ferrite core loss with nonsinusoidal waveforms using only steinmetz parameters," in *Proc. IEEE Workshop Comput. Power Electron.*, 2002, pp. 36–41.

Skye Reese (Student Member, IEEE) received the B.S. degree in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 2020 and the M.S. and Ph.D. degrees in electrical engineering from the University of Colorado, Boulder, CO, USA, in 2021 and 2024, respectively.

Her research is conducted as part of the Colorado Power Electronics Center, Department of Electrical, Computer and Energy Engineering, University of Colorado, Boulder. Her research interests include converter-level design automation and modeling methods, and the application of data science to power electronics.

Dr. Reese was the recipient of the NSF GRFP Fellowship award in 2021.

Bailey Sauter received the B.S. degree in electrical engineering from Oregon State University, Corvallis, OR, USA, in 2022 and the M.S. degree in electrical engineering from the University of Colorado, Boulder, CO, USA, in 2024.

He previously conducted research as part of the Colorado Power Electronics Center, Department of Electrical, Computer and Energy Engineering, University of Colorado, Boulder and prior to that with the Oregon State Energy Systems Laboratory.

Dragan Maksimović (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Belgrade, Belgrade, Serbia, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, CA, USA, in 1989.

From 1989 to 1992, he was with the University of Belgrade. Since 1992, he has been with the Department of Electrical, Computer and Energy Engineering, University of Colorado, Boulder, CO, USA, where he is currently a Distinguished Professor and Director of the Colorado Power Electronics Center. He has coauthored more than 350 publications and the textbooks Fundamentals of Power Electronics and Digital Control of High-Frequency Switched-Mode Power Converters. His research interests include power electronics for renewable energy sources and energy efficiency, high-frequency power conversion using wide-bandgap semiconductors, and digital control of switched-mode power converters.

Prof. Maksimović was the recipient of the 1997 National Science Foundation CAREER Award, the IEEE Power Electronics Society (IEEE PELS) Transactions Prize Paper Award in 1997, the IEEE PELS Prize Letter Awards in 2009 and 2010, the University of Colorado Inventor of the Year Award in 2006, the IEEE PELS Modeling and Control Technical Achievement Award for 2012, the Holland Excellence in Teaching Awards in 2004, 2011 and 2018, the Charles Hutchinson Memorial Teaching Award for 2012, the 2013 Boulder Faculty Assembly Excellence in Teaching Award, the 2020 College of Engineering and Applied Sciences Research Award, the 2022 IEEE PELS R. David Middlebrook Achievement Award, and the 2023 IEEE William E. Newell Power Electronics Award.