# Generative Adversarial Networks with Physics-informed Losses for High-resolution Load Profile Generation and Inpainting

Swodesh Sharma, Apeksha Ghimire, Shashwot Shrestha, Rachana Subedi, and Sushil Phuyal

***Abstract*—Accurate load profile data are essential for optimizing energy systems. However, real-world datasets often suffer from low resolution and significant missing values. To address these challenges, this paper introduces physics-informed loss generative adversarial network (PIL-GAN), a model that combines generative adversarial networks (GANs) with physics-informed losses (PILs) derived from physics-informed neural networks (PINNs) that are integrated directly into the generator. High-resolution load profiles are generated that not only fill in missing data but also ensure that the generated profiles adhere to physical laws governing the energy systems, such as energy conservation and load fluctuations. By embedding domain-specific physics into the generation process, the proposed model significantly enhances data quality and resolution for low-quality datasets. The experimental results demonstrate notable gains in data accuracy, resolution, and consistency, making PIL-GAN an effective tool for energy management systems. The PIL-GAN also has broader applicability in other fields such as generating and inpainting high-resolution datasets for energy systems, industrial processes, and any domain in which data must comply with real-world physical laws or operational requirements.**

***Index Terms*—Data inpainting, energy system, generative adversarial network (GAN), physics-informed loss (PIL), load profile, energy management system, neural network.**

## I. Introduction

THE rapid evolution of power systems hinges on the efficient sharing and application of data, ranging from educational initiatives to advanced artificial intelligence (AI) algorithms. Modeling, forecasting, and optimization for power systems require high-quality load profile data. Unfortunately, such data are sparse with low resolution, and contain missing entries owing to sensor or communication issues [1].

As most utility sectors are concerned about privacy, open-source data for load profile studies are rarely available. The available data also suffer from low resolution and significant missing values. Therefore, it is crucial to have high-quality data that facilitate more advanced research and analysis. However, utilities are reluctant to release high-resolution datasets, which may result in leakage of grid parameters, hacks on the information of customers, or the discovery of grid vulnerabilities [2], [3]. This has motivated the use of AI-based methods to tackle these challenges, compensate for data limitations, and preserve data privacy and security.

Unfortunately, traditional models for load profile generation face challenges in achieving adequate solutions. Owing to their capability to learn complex data distributions, generative adversarial networks (GANs) have become prominent tools for data generation and imputation [4]. Although GANs alone do not necessarily capture the physical laws that underly the power system, the data generated by GANs may violate basic rules such as energy conservation, peak load information, and load fluctuation owing to temperature. By explicitly incorporating physical constraints embedded in the learning process, physics-informed neural networks (PINNs) boost the capacity of physical learning and produce consistent results [5].

Combining GANs with physics-informed losses (PILs) offers a promising avenue for generating synthetic load profiles that are both realistic and physically plausible. This hybrid model addresses the issues of data sparsity and missing entries. In this paper, we propose the PIL-GAN model that combines GANs with PILs for high-resolution load profile generation and inpainting, ensuring physical consistency. The key contributions of this paper are summarized as follows.

PIL-GAN synergistically combines the generation power of GANs with the physics-informed constraints of PINNs to impute missing segments into load profiles. PIL-GAN can be utilized for super-resolution tasks, i.e., converting low-resolution load profiles into high-resolution load profiles. It can also be used to generate complete synthetic datasets using a temperature-profile reference.

We validate the PIL-GAN using a comprehensive experimental setup and demonstrate its effectiveness compared with baseline models.

The remainder of this paper is organized as follows. Section II conducts a literature review on the deep learning

Manuscript received: October 24, 2024; revised: February 23, 2025; accepted: July 17, 2025. Date of CrossCheck: July 17, 2025. Date of online publication: August 1, 2025.

The authors would like to express their gratitude to Pecan Street Inc. for providing the valuable dataset that has been essential to this research. This dataset has been utilized to advance research within our academic institution, and it has significantly contributed to our study's insights and conclusions.



S. Sharma (corresponding author), A. Ghimire, S. Shrestha, R. Subedi, and S. Phuyal are with the Institute of Engineering, Pulchowk Campus, Lalitpur, Nepal (e-mail: xarmaswodesh@gmail.com; ghimireapeksha13@gmail.com; shashwotshrestha.7@gmail.com; rachana.subedi.29@gmail.com; sushilphuyal.sp@gmail.com).

DOI: 10.35833/MPCE.2024.001153

based models for load profile imputation and forecasting, GANs, and PINNs. Section III details the methodology of the PIL-GAN for generating GAN-based load profiles with an attached PIL function, and describes the experimental setup. Section IV presents the results and discussion, and Section V concludes this paper.

## II. Literature Review

Smart and advanced metering infrastructures have resulted in an exponential growth of load data available in applications such as demand response, grid optimization, and energy forecasting. Nevertheless, owing to privacy issues and the limited availability of accessible data, it is essential to develop functions for creating synthetic load profiles that are as accurate as possible. A Lagrange interpolating polynomial method for addressing data loss in synchrophasor applications, which can also be applied to load profile imputation while balancing the accuracy and efficiency, is proposed in [6]. With increasing computational power, deep learning based models are increasingly replacing traditional methods such as *k*-nearest neighbors (KNNs) and *k*-means.

### *A. Deep Learning Based Models for Load Data Imputation and Forecasting*

Previous studies have investigated the application of machine learning models in load data imputation and forecasting tasks, with a focus on long short-term memory (LSTM) models owing to their superior performance in handling long-term dependencies in temporal data sequences. For instance, [7] studies convolutional neural network (CNN), recurrent neural network (RNN), and hybrid CNN-RNN models to forecast load demand. Local patterns are learned effectively from the Korea Power Exchange dataset, and the CNN model excels particularly in short-term forecasting.

Reference [8] proposes a context encoder model using deep CNNs, optimized with reconstruction and adversarial losses to impute missing wind farm data by capturing spatio-temporal patterns. On the other hand, in [9], the hybrid CNN-LSTM model is utilized to handle the missing data of electricity consumption in smart homes. The hybrid model uses CNN for local pattern detection and LSTM to capture long-term dependencies to reduce errors with the minimum computational cost compared with the traditional model.

Studies have also demonstrated that LSTM models are effective in energy forecasting tasks. Reference [10] compares the CNN and LSTM models for short-term load forecasting in photovoltaic plants based on data from a plant in Morocco. CNN models perform better in simpler setups, whereas LSTM models perform better in deeper architectures over longer time horizons. A multivariate time-series data imputation model using a two-step LSTM is presented in [11], which simultaneously exploits temporal and cross-dimensional correlations to achieve superior performance across different datasets.

Models with more advanced implementations that integrate multiple architectures have been shown to improve the results. For example, the attention-based bidirectional LSTM (BiLSTM) with convolution layer model suggested in [12] combines a CNN, BiLSTM, and attention mechanisms for short-term load forecasting. When applied to short-term load forecasting, the model outperforms the CNN, BiLSTM, and XGBoost. The attention mechanism allows the model to focus on relevant historical data. LSTM-based models have also been successfully applied in other domains. For example, in [13], missing data from medical records are imputed using a BiLSTM model mixed with random forest and principal component analysis, which outperforms cubic spline interpolation and KNN on hypertensive disorder records with improved accuracy.

### *B. GANs*

With the increasing demand of creating completely new and realistic data, particularly in areas such as synthetic data generation while preserving privacy, LSTMs alone may not be sufficient. Reference [14] introduces the GAN, which has become a powerful technique for generating realistic synthetic data by learning the underlying data distributions in an unsupervised manner. Two neural networks, a generator ($G$) and a discriminator ($D$), play a min-max game in a GAN. $G$ seeks to generate data identical to the real data, whereas $D$ seeks to distinguish between synthetic and real samples. The architecture of GAN is illustrated in Fig. 1.

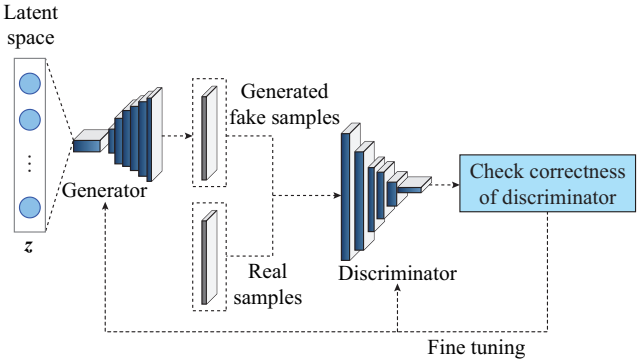


Fig. 1. Architecture of GAN.

The optimization problem is formulated as:

$$\min_{G}\max_{D}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[\ln D(\boldsymbol{x})\right]+\mathbb{E}_{z\sim p_z(z)}\left[\ln\left(1-D\left(G(\boldsymbol{z})\right)\right)\right] \quad (1)$$

where $\boldsymbol{x}$ is the actual data drawn from the true distribution $p_{\text{data}}(\boldsymbol{x})$; $\mathbb{E}$ is the expected value (average) over the corresponding distribution; and $\boldsymbol{z}$ is a latent vector sampled from the prior distribution $p_z(\boldsymbol{z})$ [15]. $G(\boldsymbol{z})$ generates data in the data space by mapping from $\boldsymbol{z}$ to minimize the difference with respect to the real data distribution, whereas $D(\boldsymbol{x})$ is designed to output the probability that a given sample is real.

Reference [16] introduces a GAN-based imputation method with a modified gated recurrent unit to capture the temporal dependencies in a multivariate time-series, validated on medical and environmental datasets. GANs have also been applied extensively in the energy sector to generate synthetic load profiles that preserve the statistical properties of real data without exposing sensitive information [17]. For instance, [18] employs a conditional GAN to generate synthetic time-series load data conditioned on factors such as season and load type, effectively capturing complex patterns in transmission-level load data. Reference [19] proposes a multi-load

GAN-based method that considers spatiotemporal correlations to generate synthetic data for groups of electricity consumers, thereby enhancing the resolution of the generated data for microgrid and distribution system studies. Furthermore, to address class imbalances and the need for data with blended characteristics, [20] introduces the blender GAN, which is a multi-target conditional GAN capable of generating synthetic data by blending multiple classes in user-defined proportions.

In comparative studies, [21] evaluates various GAN architectures to generate synthetic datasets of power systems. This study highlights the importance of model selection and hyperparameter tuning for generating high-quality synthetic data. The models are validated using statistical methods and machine learning classifiers to ensure that the synthetic data are indistinguishable from real data. GANs have also been applied to generate other types of power system data. Reference [22] utilizes a Wasserstein GAN to generate synthetic time-series data for dynamic thermal line ratings (ampacity) and electricity pool prices, addressing the limited availability of such data for training deep learning models in smart grids. In the realm of phasor measurement unit (PMU) data, [23] uses GANs to generate synthetic dynamic PMU data, providing a model-free method to produce realistic time-series data crucial for monitoring power systems while overcoming privacy constraints.

Similarly, [24] introduces the profile super-resolution GAN, which is a two-stage GAN model for upsampling low-resolution load profiles into high-resolution profiles. The model restores high-frequency components in the first stage and eliminates unrealistic fluctuations in the second stage, outperforming other models in terms of shape-based metrics and non-intrusive load monitoring. SparseGAN, as described in [25], addresses time-series generation challenges by incorporating sparse self-attention within the GAN architecture, allowing the model to capture long-range dependencies more effectively. Its performance is validated using both synthetic and real-world datasets.

A privacy-preserving method for generating customer load profiles using an information maximizing GAN and multivariate kernel density estimation is presented in [26]. This method maintains privacy while generating realistic data that capture uncertainties in electricity consumption. In [27], a comparative analysis of energy load forecasting techniques highlights the superiority of hybrid models, particularly those combining LSTM and CNN. GAN-based data augmentation further enhances the performance, particularly with limited training data, which is proven to be effective for load forecasting. Finally, [28] proposes the load profile inpainting network, which is a GAN-based model to restore missing segments of load data during demand response events. It features a two-stage generator: a coarse network for initial estimates, and a fine-tuning network using self-attention and gated convolution layers for refinement. The innovative loss functions improve the precision, yielding a 15%-30% improvement over existing models. A recent study in [29] has classified deep learning based multivariate imputation methods into predictive, generative, and large-model-based categories, providing a detailed comparison of their advantages and disadvantages. This study focuses on enhancing GAN-based imputation with PIL and comparing it with some predictive models, but a broader analysis remains outside of its scope. Time-series GAN (TimeGAN), which is introduced in [30] and shown in Fig. 2, combines unsupervised adversarial loss with supervised learning signals, and preserves time-based correlations to address the gap in capturing temporal dependencies across sequential steps. Instead of depending only on adversarial feedback, TimeGAN improves the temporal fidelity in generated sequences by incorporating a step-wise supervised loss, which encourages the model to learn transition dynamics. Furthermore, time-series data can be mapped into a lower-dimensional latent space using an embedding network, thereby increasing the training efficiency and stability. Evaluations on synthetic and real-world datasets have shown that TimeGAN consistently outperforms existing benchmark models in terms of fidelity and forecasting quality.

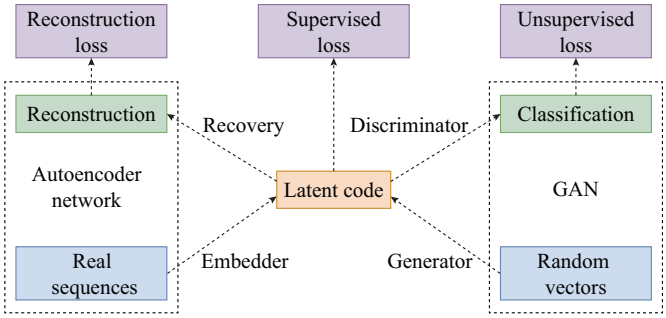


Fig. 2. Architecture of TimeGAN.

### C. PINNs

Although GANs have been successful in generating synthetic data, there is a significant challenge in generating synthetic data that follow the physical constraints of a power system. PINNs, as described in [31], combine machine learning with physical constraints represented by a set of partial differential equations, as shown in Fig. 3. In Fig. 3, $x$ and $t$ represent different independent variables fed into the neural network. The hidden layers with $\sigma$ denote activation functions, while $u$, $v$, $p$, and $\phi$ are the predicted physical quantities of interest. Automatic differentiation (AutoDiff) computes their derivatives, which are incorporated into the loss function $\mathcal{L}$ for optimization. Rather than relying solely on data-driven training, PINNs use the governing equations as regularizers in the loss function $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{data}} + \lambda \mathcal{L}_{\text{physics}} \tag{2}$$

where $\mathcal{L}_{\text{data}}$ is the conventional data loss; $\mathcal{L}_{\text{physics}}$ is the physical loss that enforces compliance with physical laws; and $\lambda$ is a weighting parameter.

PINNs have been applied in power systems for parameter estimation and predictive maintenance. For instance, [32] proposes a physics-informed machine learning method for parameter estimation of DC-DC converters, combining machine learning with physical models to estimate parameters such as capacitance and inductance with high accuracy. As

highlighted in a comprehensive review [33], the integration of machine learning with physical models holds significant potential for improving the generalizability and performance of predictive maintenance strategies for power converters, concurrently reducing their data requirements.

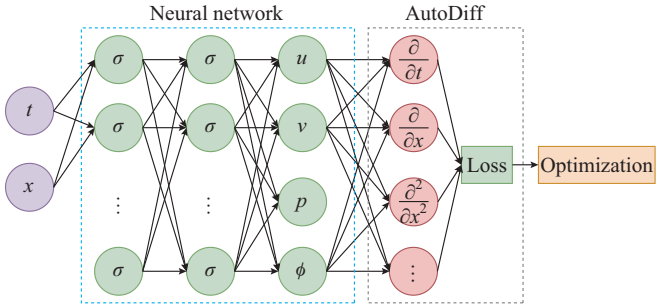


Fig. 3. Architecture of PINN.

### D. Research Gap and PIL-GAN

The combination of GANs and physics-informed constraints for creating synthetic load profiles has not been explored extensively. Previous studies have either focused on the production of realistic data without imposing strict physical constraints, or introduced physical laws that do not fully explore the data generation capabilities of GANs. This research gap leads to the development of a unique method that incorporates the best aspect of GANs in working with complicated data distributions while ensuring physical plausibility from physics-informed constraints similar to PINNs. In this paper, the PIL-GAN ensures that the synthetic data possess statistical characteristics that closely resemble the real load profile data, while also satisfying physics-based constraints, thus improving the usefulness of synthetic data in power systems.

## III. Methodology

### A. Overview

To address both high-resolution load profile generation and inpainting tasks, the PIL-GAN is introduced in this paper. This involves adopting the generative properties of GANs to generate load profiles while imposing physical constraints that enforce physical accuracy and conformity with known physical laws and operating rules.

### B. Data Description and Preprocessing

The data used in this paper consist of electricity consumption data with 15 min interval from 25 residential houses collected in 2018 in Austin, USA, obtained from the Pecan Street dataset [34]. The two main columns of interest are the net electricity consumed by the grid and solar generation.

For the inpainting task, the total energy consumption is calculated by combining the net electricity consumed by the grid and solar generation. A positive value in the consumption indicates that the house consumes electricity from the grid, whereas a negative value indicates that surplus solar energy is being injected back into the grid. Using this information, the total consumption of each house in each 15 min interval is computed.

Subsequently, the energy consumption of all 25 houses is summed to generate a smooth aggregate curve. This results in an annual energy consumption profile with 15 min intervals, which is used for both training and validation purposes. The typical load profile of the combined datasets illustrates the aggregate load fluctuations of all houses over a 24 hour period, as shown in Fig. 4.

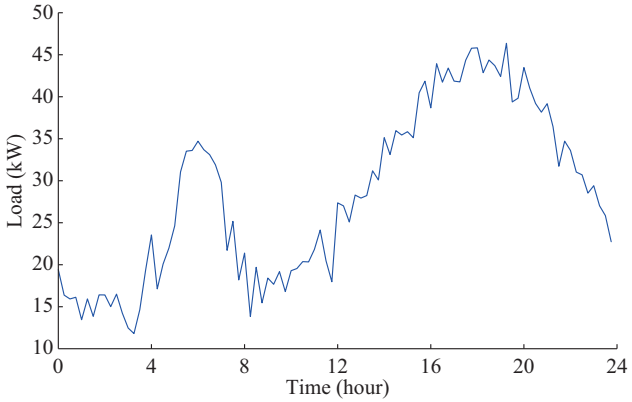


Fig. 4. Typical load profile of combined datasets.

Before training, the load profiles are normalized using min-max normalization as:

$$\boldsymbol{x}_{\text{norm}}(t) = \frac{\boldsymbol{x}(t) - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})} \tag{3}$$

where $\boldsymbol{x}(t)$ is the load value at time $t$; and $\boldsymbol{x}_{\text{norm}}(t)$ is the normalized load at time $t$.

### C. Architecture of PIL-GAN

#### 1) Generator

The generator is designed to complete the missing pieces from the load profiles without violating the physics-based constraints. The masked load profile is generated through an element-wise product of the load profile and a randomly generated mask of the same size, where 1 indicates the presence of data and 0 represents the missing data. The inputs to the generator include the masked load profiles, the temperature profiles of the day, and the mask. The generator includes several fully connected layers with rectified linear unit (ReLU) as the activation for all the hidden layers. The final output is passed through ReLU, which is appropriate because the target load profiles are non-negative. This simple architecture sufficiently proves the applicability of the proposed model and can be considered a starting point for further studies on additional and more complex constructions.

#### 2) Discriminator

The discriminator obtains a complete load profile, which is both real and generated, along with the temperature profile. It provides a measure of the probability that the load profile is authentic. Similar to the generator, the discriminator also contains fully connected layers that use ReLU for the hidden layers and a sigmoid function in the output layer, as the discriminator predicts the probability of real and fake data between 0 and 1. Thus, by maintaining a simple and uniform architecture, we aim to convey the essence of applying the PIL-GAN without further complications.

For both the generator and discriminator, ReLU is chosen as the activation function in all the hidden layers to ensure that the outputs remain positive, aligning with the nature of the load profile data. The overall architecture of PIL-GAN is illustrated in Fig. 5, demonstrating the inclusion of temperature profiles.

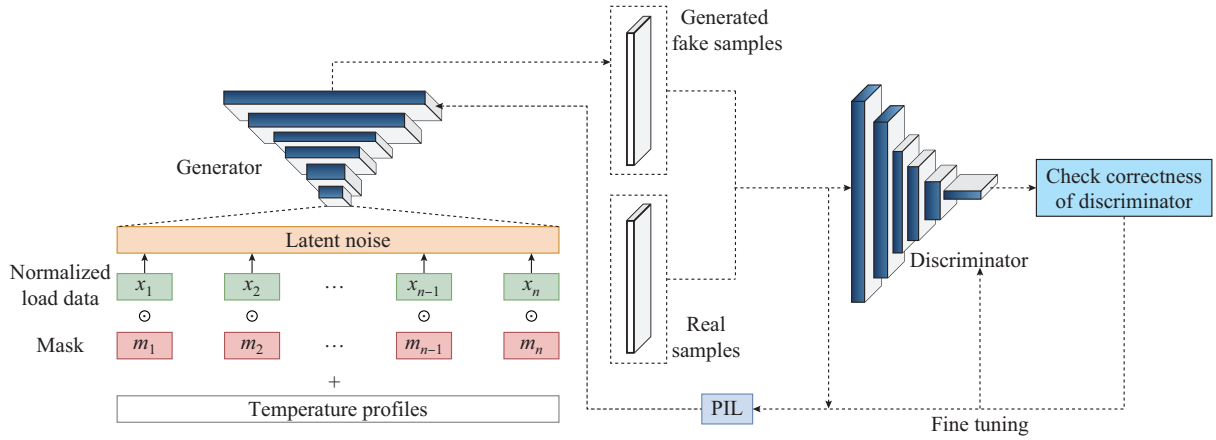


Fig. 5. Overall architecture of PIL-GAN.

### D. PIL Functions

We incorporate a series of PIL functions to ensure that the generated load profiles satisfy the real-world physical constraints. These PIL functions enforce realistic behaviors such as energy conservation, limited load fluctuations, peak load constraints, and smooth temporal transitions. The parameters governing these constraints are extracted from historical data including temperature and load profiles, which are prior to training and remain fixed during the process. For training, the temperature profiles are analyzed using historical data with similar temperature patterns.

#### 1) Energy Conservation Loss

The total energy consumed in a day must not exceed the maximum allowable energy $E_{\max}$, which is obtained from historical data. The energy conservation loss $\mathcal{L}_{\text{energy}}$ penalizes load profiles that exceed this limit.

$$\mathcal{L}_{\text{energy}} = \frac{1}{N}\sum_{i=1}^{N}\max\left(0, \sum_{t=1}^{T}\hat{\boldsymbol{x}}_{i,t} - E_{\max}\right) \tag{4}$$

where $N$ is the batch size; $T$ is the total number of time steps per day; and $\hat{\boldsymbol{x}}_{i,t}$ is the generated load at time $t$ for sample $i$.

#### 2) Load Fluctuation Loss

To avoid unrealistic spikes or drops in load between consecutive time steps, we introduce a maximum allowable change $\Delta_{\max}$ based on historical data. The load fluctuation loss $\mathcal{L}_{\text{fluctuation}}$ penalizes the load profiles when the change between consecutive time steps exceeds this threshold.

$$\mathcal{L}_{\text{fluctuation}} = \frac{1}{N(T-1)}\sum_{i=1}^{N}\sum_{t=1}^{T-1}\max\left(0, \left|\hat{\boldsymbol{x}}_{i,t+1} - \hat{\boldsymbol{x}}_{i,t}\right| - \Delta_{\max}\right) \tag{5}$$

#### 3) Peak Load Constraint

The load at any given time should not exceed the historical maximum peak load $P_{\max}$. The peak load loss $\mathcal{L}_{\text{peak}}$ penalizes the generated load profiles violating the following constraint:

$$\mathcal{L}_{\text{peak}} = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\max\left(0, \hat{\boldsymbol{x}}_{i,t} - P_{\max}\right) \tag{6}$$

#### 4) Temporal Smoothness Loss

To reflect the gradual changes observed in the real load profiles, large swings should not occur within a given time window. The temporal smoothness loss $\mathcal{L}_{\text{smoothness}}$ penalizes the maximum deviation inside each sliding window of length $w$:

$$\mathcal{L}_{\text{smoothness}} = \frac{1}{N(T-w)}\sum_{i=1}^{N}\sum_{t=1}^{T-w}\max_{\tau=1,2,\ldots,w}\left|\hat{\boldsymbol{x}}_{i,t+\tau} - \hat{\boldsymbol{x}}_{i,t}\right| \tag{7}$$

By considering the maximum absolute difference over the window, we capture large deviations that could occur anywhere within that window, thereby enforcing smoother transitions over short timespans.

#### 5) Total Loss

The total loss $\mathcal{L}_{\text{PINN}}$ during training is the weighted sum of the individual PIL functions. These functions ensure that the generated load profiles are realistic and adhere to the physical constraints derived from the historical data.

$$\mathcal{L}_{\text{PINN}} = \lambda_{\text{energy}}\mathcal{L}_{\text{energy}} + \lambda_{\text{fluctuation}}\mathcal{L}_{\text{fluctuation}} + \lambda_{\text{peak}}\mathcal{L}_{\text{peak}} + \lambda_{\text{smoothness}}\mathcal{L}_{\text{smoothness}} \tag{8}$$

where $\lambda_{\text{energy}}$, $\lambda_{\text{fluctuation}}$, $\lambda_{\text{peak}}$, and $\lambda_{\text{smoothness}}$ are the hyperparameters that balance the contribution of each loss term. For simplicity, they are initially set to be 1 as the starting point, with future tuning planned for optimization.

Although the physical constraints in this paper remain simple, they significantly improve the quality of the imputed load profiles by ensuring realistic energy consumption patterns, smooth transitions, and adherence to the observed limits. Moreover, the application of the PIL-GAN is not limited to load profile imputation. It can be extended to more complex tasks where physical loss functions or constraints are derived from complex differential equations governing system dynamics. This makes the PIL-GAN highly adaptable to a wide range of applications that require synthetic data generation, while maintaining fidelity to physical laws.

### E. Training Procedure

#### 1) Adversarial Training

Adversarial training is employed to optimize the generator

and discriminator. The discriminator is trained to maximize the probability of correctly identifying real and generated load profiles, whereas the generator aims to generate load profiles that the discriminator cannot distinguish from the real ones.

The discriminator loss $\mathcal{L}_D$ is given by:

$$\mathcal{L}_D = -\frac{1}{N}\sum_{i=1}^{N}\left(\ln D\left(\boldsymbol{x}_i\right) + \ln\left(1 - D\left(\hat{\boldsymbol{x}}_i\right)\right)\right) \tag{9}$$

where $\boldsymbol{x}_i$ is the real load profile; and $\hat{\boldsymbol{x}}_i$ is the generated load profile.

*2) Generator Loss with Physics-informed Constraints*

The generator loss consists of two components, i.e., adversarial loss $\mathcal{L}_{\text{adv}}$ and total loss $\mathcal{L}_{\text{PINN}}$.

$$\mathcal{L}_G = \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{PINN}} \tag{10}$$

$$\mathcal{L}_{\text{adv}} = -\frac{1}{N}\sum_{i=1}^{N}\ln D\left(\hat{\boldsymbol{x}}_i\right) \tag{11}$$

*3) Optimization*

We use the Adam optimizer for the generator and discriminator, with the learning rates $\alpha_G$ and $\alpha_D$, respectively. The training procedure is shown in Algorithm 1 by iteratively updating the discriminator and generator.

### F. Inpainting Procedure

Once the PIL-GAN is trained, a generator is used to inpaint the missing segments in the new load profiles.

Given a masked load profile vector $\boldsymbol{x}$, mask $\boldsymbol{m}$, and temperature $\boldsymbol{t}$, the generator produces an inpainted profile $\hat{\boldsymbol{x}}$. The observed and generated data are combined to form a completely inpainted profile $\boldsymbol{x}_{\text{inpainted}}$:

**Algorithm 1**: training procedure for PIL-GAN

1: Discriminator update:
2: Sample real profiles $\{\boldsymbol{x}_i\}_{i=1}^{N}$
3: Generate inpainted profiles $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{N}$
4: Compute $\mathcal{L}_D$ and update discriminator
5: Generator update:
6: Generate inpainted profiles $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{N}$
7: Compute $\mathcal{L}_G = \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{PINN}}$
8: Update generator

$$\boldsymbol{x}_{\text{inpainted}} = \boldsymbol{m} \odot \boldsymbol{x} + (\mathbf{1} - \boldsymbol{m}) \odot \hat{\boldsymbol{x}} \tag{12}$$

where $\odot$ denotes element-wise multiplication; and $\mathbf{1}$ is a vector of ones.

### G. Denormalization and Post-processing

After inpainting, the load profiles are denormalized to return to the original scale:

$$\boldsymbol{x}_{\text{denorm}} = \hat{\boldsymbol{x}}\left(x_{\max} - x_{\min}\right) + x_{\min} \tag{13}$$

where $\boldsymbol{x}_{\text{denorm}}$ is the denormalized load profile vector; $\hat{\boldsymbol{x}}$ is the normalized (predicted) load profile vector; and $x_{\max}$ and $x_{\min}$ are the scalar maximum and minimum values of the original load profile, respectively. The operations are applied element-wise, with scalars broadcast across the vector dimension.

### H. Visualization and Evaluation

To assess the performance of the PIL-GAN, we evaluate it for two main tasks: high-resolution load profile generation and inpainting. The inpainting tasks include both peak inpainting and valley inpainting, as well as inpainting in missing segments. These tasks are illustrated in Fig. 6.

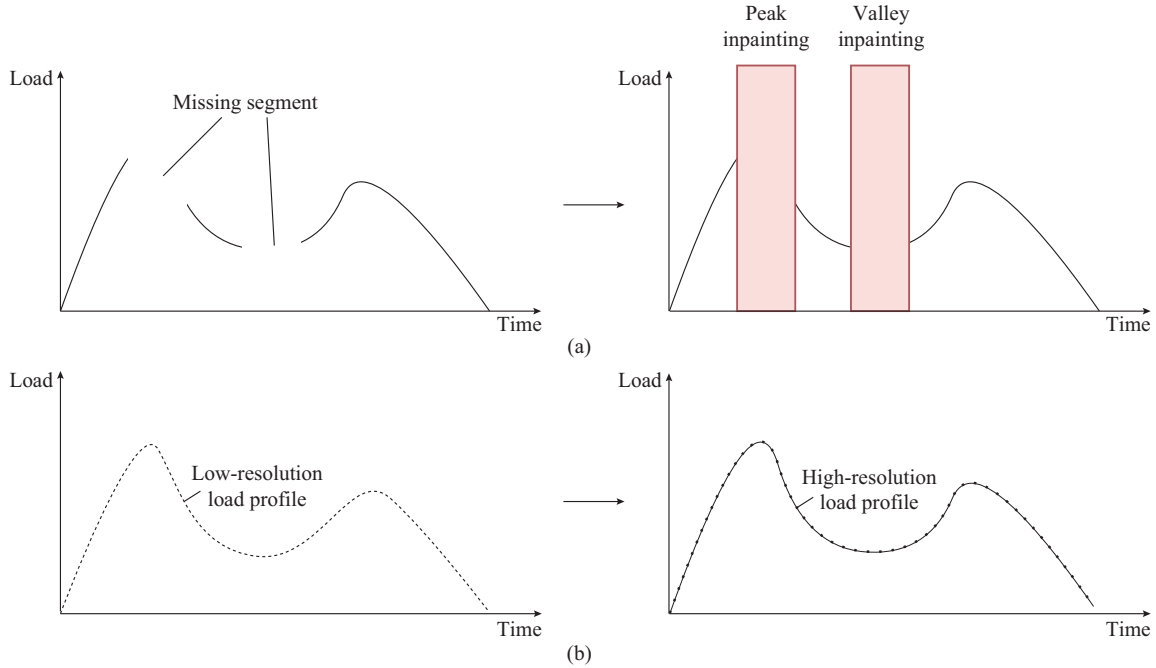


Fig. 6. Illustration of inpainting tasks. (a) Peak and valley inpainting. (b) High-resolution load profile generation.

The original, masked, and inpainted load profiles are visualized to qualitatively assess the performance of the PIL-GAN. Quantitative evaluation metrics are computed between the inpainted segments and the ground truth to evaluate the accuracy. These metrics include the mean absolute percentage error (MAPE), root mean square error (RMSE) (as a per-

centage), peak discrepancy error (PDE), trough discrepancy error (TDE), energy deviation percentage (EDP), and spectral distortion percentage (SDP).

*1) MAPE*

The MAPE is calculated as:

$$MAPE=\frac{1}{|\mathcal{M}|}\sum_{t\in\mathcal{M}}\left|\frac{\boldsymbol{x}_{\text{inpainted}}(t)-\boldsymbol{x}_{\text{true}}(t)}{\boldsymbol{x}_{\text{true}}(t)+\varepsilon}\right|\times 100\% \quad (14)$$

where $\mathcal{M}$ is the set of time steps corresponding to the missing segments; $\boldsymbol{x}_{\text{true}}(t)$ is the original load value at time $t\in\mathcal{M}$; and $\varepsilon$ is a small constant to prevent division by zero.

*2) RMSE*

The RMSE is calculated as:

$$RMSE=\sqrt{\frac{1}{|\mathcal{M}|}\sum_{t\in\mathcal{M}}\left(\boldsymbol{x}_{\text{inpainted}}(t)-\boldsymbol{x}_{\text{true}}(t)\right)^2} \quad (15)$$

And RMSE as a percentage is given by:

$$RMSE=\frac{RMSE}{\frac{1}{|\mathcal{M}|}\sum_{t\in\mathcal{M}}\left|\boldsymbol{x}_{\text{true}}(t)\right|}\times 100\% \quad (16)$$

*3) PDE*

The PDE is calculated as:

$$PDE=\frac{\left|\max\limits_{t\in\mathcal{P}}\boldsymbol{x}_{\text{inpainted}}(t)-\max\limits_{t\in\mathcal{P}}\boldsymbol{x}_{\text{true}}(t)\right|}{\left|\max\limits_{t\in\mathcal{P}}\boldsymbol{x}_{\text{true}}(t)\right|}\times 100\% \quad (17)$$

where $\mathcal{P}$ is the peak period within $\mathcal{M}$.

*4) TDE*

Similarly, the TDE is calculated as:

$$TDE=\frac{\left|\min\limits_{t\in\mathcal{T}}\boldsymbol{x}_{\text{inpainted}}(t)-\min\limits_{t\in\mathcal{T}}\boldsymbol{x}_{\text{true}}(t)\right|}{\left|\min\limits_{t\in\mathcal{T}}\boldsymbol{x}_{\text{true}}(t)\right|}\times 100\% \quad (18)$$

where $\mathcal{T}$ is the trough period within $\mathcal{M}$.

*5) EDP*

The EDP is calculated as:

$$EDP=\frac{\left|\sum\limits_{t\in\mathcal{M}}\boldsymbol{x}_{\text{inpainted}}(t)-\sum\limits_{t\in\mathcal{M}}\boldsymbol{x}_{\text{true}}(t)\right|}{\left|\sum\limits_{t\in\mathcal{M}}\boldsymbol{x}_{\text{true}}(t)\right|}\times 100\% \quad (19)$$

*6) SDP*

The SDP is calculated as:

$$SDP=\frac{\sqrt{\frac{1}{N_{\text{f}}}\sum\limits_{k=1}^{N}\left(\left|\boldsymbol{x}_{\text{inpainted}}(k)\right|-\left|\boldsymbol{x}_{\text{true}}(k)\right|\right)^2}}{\frac{1}{N_{\text{f}}}\sum\limits_{k=1}^{N}\left|\boldsymbol{x}_{\text{true}}(k)\right|}\times 100\% \quad (20)$$

where $\boldsymbol{x}_{\text{inpainted}}(k)$ and $\boldsymbol{x}_{\text{true}}(k)$ are the discrete Fourier transforms of the inpainted and true signals over $\mathcal{M}$, respectively; and $N_{\text{f}}$ is the number of frequency components in load curve.

## I. Model Complexity and Training Efficiency

In addition to GAN and PIL-GAN, the LSTM, RNN, and variational autoencoder (VAE) are also employed for comparative analysis. The structures of the LSTM, RNN, and VAE are shown in Figs. 7-9, respectively. The baseline benchmark model also maintains a cubic spline interpolation. Note that in Fig. 7, $C_t$ is the cell state; $f_t$ is the forget gate activation; $i_t$ is the input gate activation; $O_t$ is the output gate activation; $h_t$ is the hidden state; *sigmoid* is the sigmoid function; and tanh is the hyperbolic tangent function. In Fig. 8, $L$ is the output; $X$ is the input; and $U$, $V$, and $W$ are the input-to-hidden, hidden-to-hidden, and hidden-to-output weights, respectively. In Fig. 9, $X$ is the input to the encoder; $C$ is the latent space representation; and $\hat{X}$ is the output produced by the decoder.

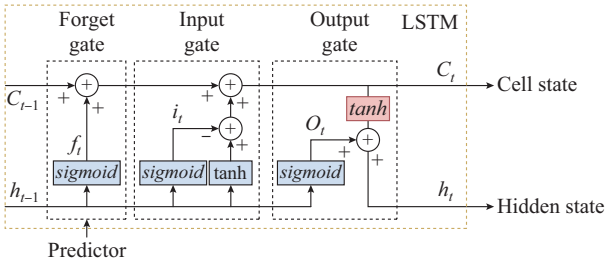


Fig. 7. Structure of LSTM.

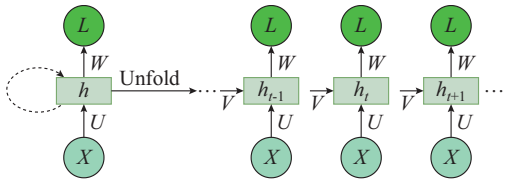


Fig. 8. Structure of RNN.

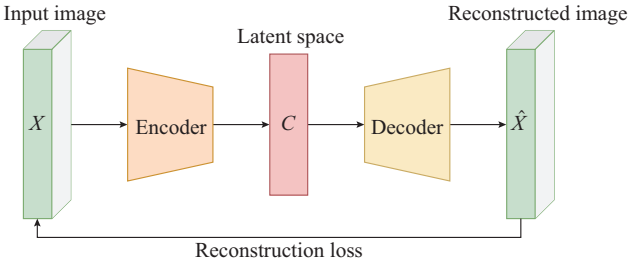


Fig. 9. Structure of VAE.

As the original data of the load profile are in 15 min intervals and are discrete, only sampled data from the cubic spline interpolation are used for inpainting instead of the whole continuous load profile, as shown in Fig. 10.

A grid search is conducted over multiple learning rates, and the hyperparameters listed in Table I represent the optimal settings at which all the models achieve their best performances.

The PIL-GAN has been compared with TimeGAN, the state-of-the-art (SOTA) model for time-series inpainting. As the task involves inpainting, and there are no explicitly defined physical differential equations, the PIL-GAN slightly underperforms TimeGAN. However, owing to its simpler architecture and shorter training time, the PIL-GAN still achieves results close to those of TimeGAN. Using the same training conditions, the generator of the PIL-GAN contains significantly fewer trainable parameters than traditional

RNNs such as LSTM, RNN, and VAE. The reduction in complexity not only accelerates the training process, but also enables higher accuracy in the generated load profiles.

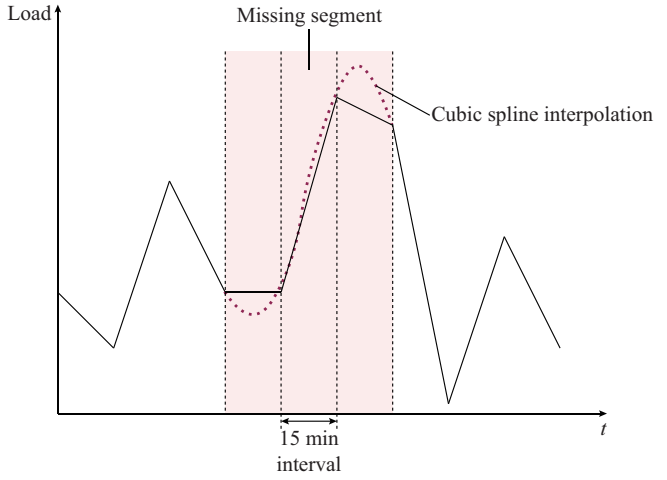


Fig. 10. Cubic spline interpolation for inpainting.

TABLE I
HYPERPARAMETERS

| Hyperparameter | Value |
|---|---|
| Number of epochs | 20000 |
| Batch size | 128 |
| Learning rate | 0.0001 |

Table II presents the number of trainable parameters and training time per epoch for each model. The GAN (both PIL-GAN and GAN (no physics) has approximately 391089 trainable parameters, which are less than half of those in the LSTM and RNN. Specifically, the LSTM has 901168, VAE has approximately 680000, and the RNN model has 862256 trainable parameters. Similarly, TimeGAN, with 400158 trainable parameters, is used for the comparison.

TABLE II
NUMBER OF TRAINABLE PARAMETERS AND TRAINING TIME PER EPOCH

| Model | Number of trainable parameters | Time per epoch (s) |
|---|---|---|
| GAN | 391089 | 0.42 |
| TimeGAN | 400158 | 1.20 |
| LSTM | 901168 | 0.68 |
| RNN | 862256 | 0.64 |
| VAE | 680000 | 0.50 |

The training of GAN is faster than sequential recurrent networks owing to fewer parameters, lower computational costs, and its inherently parallelizable architecture. The training of GAN is even faster than that of TimeGAN, because the use of RNNs in TimeGAN cannot be efficiently parallelized. Furthermore, in the quantitative analysis, it is shown that the simpler architecture of PIL-GAN does not compromise the accuracy compared with LSTM, RNN, and VAE that have more complex structures. Even with a shorter training time, its output is almost comparable to that of TimeGAN.

By incorporating PIL functions into the training procedure, the generator learns to simultaneously output data that comply with energy conservation laws, operational constraints, and temporal dynamics, which are as realistic as possible. The PIL-GAN is therefore a powerful tool for high-resolution load profile generation and inpainting.

## IV. RESULTS AND DISCUSSION

### A. Results of Inpainting

The inpainting performance of the PIL-GAN is evaluated against various baseline models such as GAN (no physics), LSTM, RNN, VAE, and cubic spline interpolation. Figure 11 shows the results of peak and valley inpainting of missing data. In both cases, 10 out of the 96 data points of the original load profile are removed around these regions (pink areas in Fig. 11), and the performances of all the models are compared.

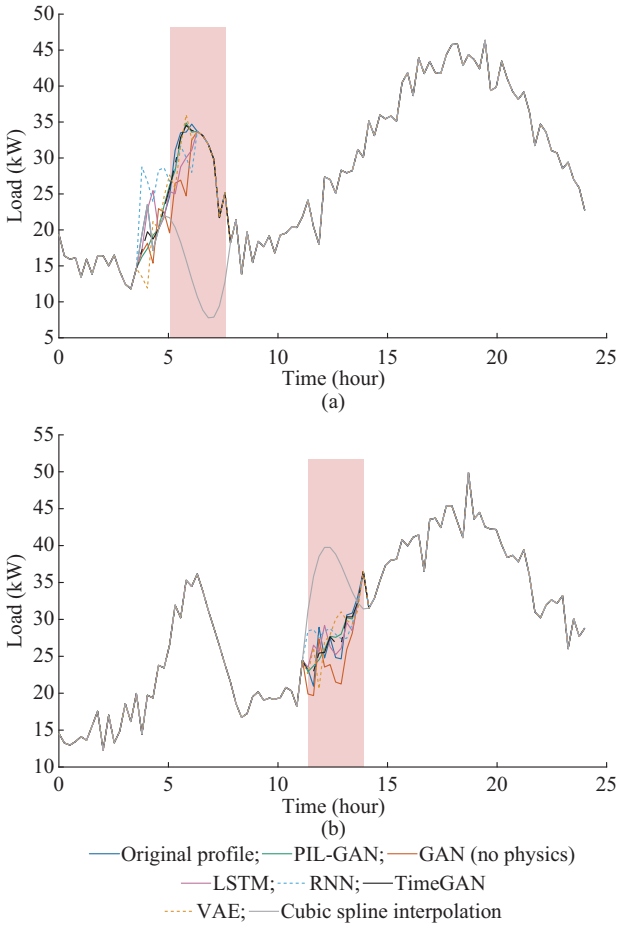


Fig. 11. Results of peak and valley inpainting of missing data. (a) Peak inpainting. (b) Valley inpainting.

The closest approximation to the original load profile is provided by TimeGAN, which effectively fills in the missing segments while maintaining physical consistency, followed by the PIL-GAN. Meanwhile, models such as LSTM, RNN, VAE, and cubic spline interpolation struggle to maintain load fluctuations and exhibit noticeable deviations in key re-

gions. The PIL-GAN performs similar to TimeGAN, while outperforming VAE and cubic spline interpolation.

### B. Results of High-resolution Load Profile Generation

The PIL-GAN is evaluated for a task in which low-resolution time-series data with few measurements are converted into high-resolution data. The results for conversion of a 20% sampled load profile to high-resolution load profile are shown in Fig. 12, which shows that PIL-GAN, compared with other models such as LSTM, RNN, VAE, and GAN (no physics), generates a much smoother high-resolution profile that closely tracks the original measurements. The smooth and realistic profiles generated by the PIL-GAN are due to physical constraints such as energy conservation. As the sampling rate decreases, the performance of the cubic spline interpolation deteriorates, as sparse data limit the ability of capturing signal variations, which is illustrated in Fig. 13.

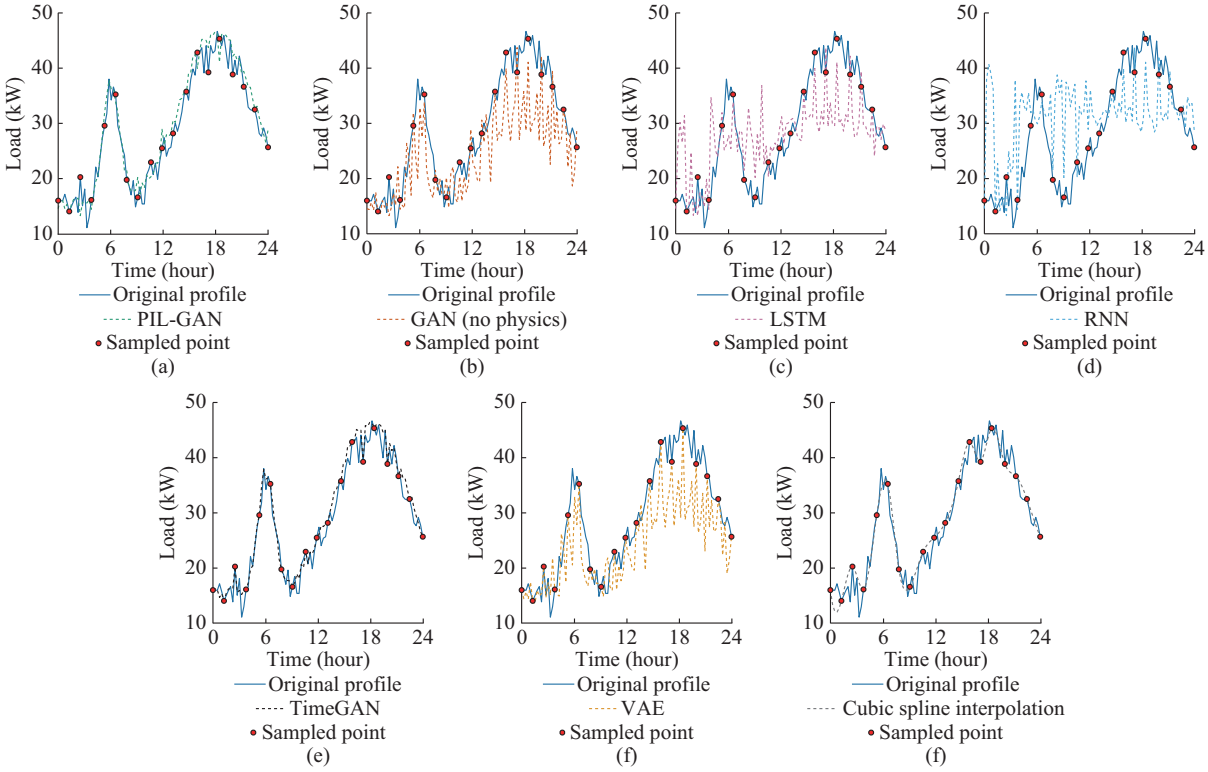


Fig. 12. Results for conversion of a 20% sampled load profile to high-resolution load profile. (a) PIL-GAN. (b) GAN (no physics). (c) LSTM. (d) RNN. (e) TimeGAN. (f) VAE. (g) Cubic spline interpolation.

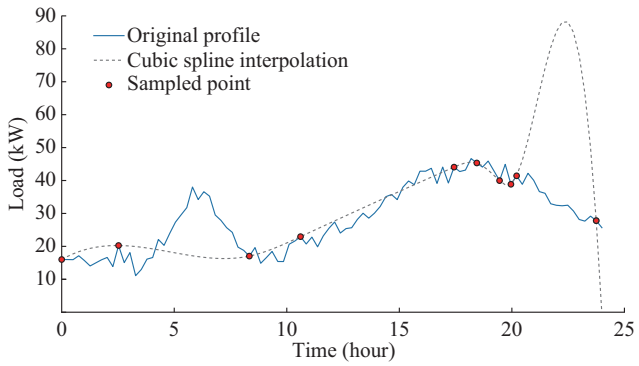


Fig. 13. Performance of cubic spline interpolation at low sampling rates.

### C. Full Synthetic Load Profile Generation

This is an application of the PIL-GAN to generate full synthetic load profiles with temperature profiles as input. As illustrated in Fig. 14, the results generated by PIL-GAN are nearly indistinguishable from the original load profile, which effectively captures the temporal patterns, peak load periods, and overall trends. However, models such as the LSTM and RNN show notable deviations, mainly during peak load periods. A cubic spline could not be used, because the load profiles are generated by feeding the temperature profiles. This demonstrates that the PIL-GAN can be applied to generate realistic load profiles that preserve key characteristics such as peak values and fluctuations based on temperature inputs.

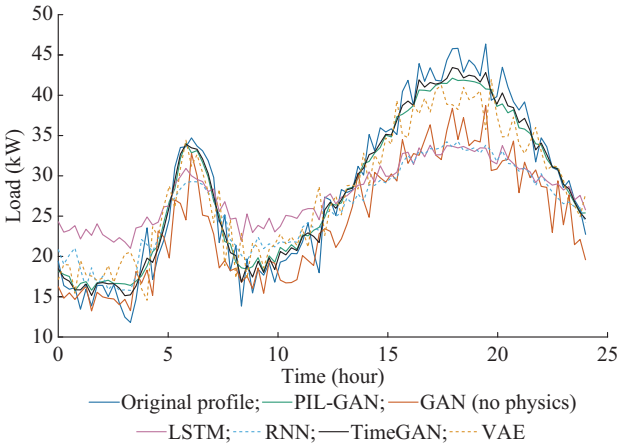


Fig. 14. Comparison of full synthetic profile generation.

### *D. Quantitative Analysis*

Tables III and IV present the quantitative evaluations of the peak and valley inpainting task and high-resolution load profile generation task of different models, respectively. We use the mean percentage error (MPE), RMSE, EDP, and SDP to comprehensively assess the performances of different models.

Table III shows that the PIL-GAN consistently outperforms the other models across all metrics in the peak and valley inpainting tasks. The PIL-GAN reduces the RMSE and MPE significantly compared with GAN (no physics). Similarly, Table V shows the comparison of inpainting performances for different missing ratios, where the error metric is the sum of absolute errors of all error metrics presented.

The ablation study on the PILs in Table VI combining peak and valley regions further confirms that the inclusion of PILs significantly improves the accuracy. Removing any of these components results in substantial performance degradation, as indicated by the increased values of MAE and RMSE.

In Table IV, a similar trend can be observed for high-resolution load profile generation task. The PIL-GAN outperforms traditional models such as LSTM, RNN, VAE, and cubic spline interpolation. In addition, PIL-GAN achieves a performance comparable to that of TimeGAN. Thus, the PIL-GAN is more effective for generating accurate high-resolution load profiles, particularly in complex nonlinear scenarios.

TABLE III
QUANTITATIVE EVALUATION OF PEAK AND VALLEY INPAINTING TASK OF DIFFERENT MODELS

| Region | Model | MAPE | RMSE (%) | PDE | TDE | EDP | SDP |
|---|---|---|---|---|---|---|---|
| Peak | TimeGAN (SOTA) | 0.6219 | 1.9696 | 0.5297 | | 0.1735 | 5.813 |
| | PIL-GAN | 0.8702 | 2.8483 | 0.4683 | | 0.2694 | 8.0778 |
| | GAN (no physics) | 2.5143 | 8.5123 | 2.7695 | | 1.9512 | 22.4167 |
| | LSTM | 2.6189 | 8.9874 | 2.9781 | | 2.0817 | 23.6238 |
| | RNN | 2.9972 | 9.8341 | 3.3124 | | 2.3184 | 25.9993 |
| | VAE | 1.9458 | 6.8834 | 2.2542 | | 1.5832 | 18.2997 |
| | Cubic spline interpolation | 3.9911 | 12.4421 | 4.0199 | | 3.4429 | 36.2078 |
| | Error reduction (excluding SOTA) | 1.0756 | 4.0351 | 1.7859 | | 1.3138 | 10.2219 |
| Valley | TimeGAN (SOTA) | 0.4939 | 1.8059 | | 8.8329 | 0.0629 | 5.1011 |
| | PIL-GAN | 0.6653 | 2.3769 | | 8.9834 | 0.1157 | 6.5968 |
| | GAN (no physics) | 1.6247 | 5.1312 | | 14.8732 | 1.4867 | 14.2425 |
| | LSTM | 1.7123 | 5.7654 | | 15.4987 | 1.6143 | 15.5312 |
| | RNN | 2.1334 | 6.6232 | | 17.2168 | 1.9222 | 17.3014 |
| | VAE | 1.4531 | 4.3898 | | 11.3813 | 1.1512 | 11.6893 |
| | Cubic spline interpolation | 3.8099 | 10.7343 | | 23.4924 | 2.9732 | 30.5987 |
| | Error reduction (excluding SOTA) | 0.7878 | 2.0129 | | 2.3979 | 1.0355 | 5.0925 |

TABLE IV
QUANTITATIVE EVALUATION OF HIGH-RESOLUTION LOAD PROFILE GENERATION TASK OF DIFFERENT MODELS

| Model | MAPE | RMSE (%) | PDE | TDE | EDP | SDP |
|---|---|---|---|---|---|---|
| TimeGAN (SOTA) | 5.210 | 6.813 | 4.970 | 9.505 | 3.080 | 22.890 |
| PIL-GAN | 6.012 | 7.421 | 4.881 | 8.928 | 3.411 | 24.110 |
| GAN (no physics) | 12.876 | 17.238 | 6.122 | 14.902 | 7.112 | 63.542 |
| LSTM | 15.991 | 20.659 | 6.289 | 16.372 | 5.212 | 57.487 |
| RNN | 18.574 | 23.877 | 6.931 | 18.562 | 6.104 | 71.209 |
| VAE | 9.341 | 12.928 | 5.211 | 11.885 | 4.408 | 32.584 |
| Cubic spline interpolation | 8.212 | 9.113 | 5.612 | 9.905 | 4.802 | 34.879 |
| Error reduction (excluding SOTA) | 2.200 | 1.692 | 0.330 | 0.977 | 0.997 | 8.474 |

Additionally, the error distributions of the peak and valley inpainting tasks for different models are illustrated in Fig. 15. The TimeGAN maintains the lowest error distribution, followed by the PIL-GAN, for both profiles, further demonstrating the robustness of incorporating physics-based constraints into the model. Similarly, the probability distributions of the predicted and true values for the different models are shown in Fig. 16. It can be observed that the PIL-GAN model produces distributions that are closer to the original load profile compared with the other models.

### *E. Discussion*

Comprehensive qualitative and quantitative analyses illustrate that the PIL-GAN significantly outperforms traditional GAN models, VAE, LSTM, and RNN, in high-resolution load profile generation and inpainting.

TABLE V
COMPARISON OF INPAINTING PERFORMANCES FOR DIFFERENT MISSING RATIOS

| Missing ratio (%) | Error | | | | | | |
|---|---|---|---|---|---|---|---|
| | TimeGAN | PIL-GAN | GAN (no physics) | LSTM | RNN | VAE | Cubic spline interpolation |
| 20 | 29.2668 | 34.6278 | 82.1323 | 86.3215 | 97.0367 | 63.3958 | 78.3634 |
| 30 | 33.1337 | 37.9836 | 88.7422 | 92.9864 | 105.0734 | 65.5560 | 83.3611 |
| 40 | 37.0005 | 41.3395 | 95.3522 | 98.7986 | 113.1101 | 67.7162 | 88.3589 |
| 50 | 40.8674 | 44.6954 | 101.9621 | 104.0560 | 121.1469 | 69.8764 | 105.3567 |
| 60 | 44.7342 | 48.0513 | 108.5721 | 110.2509 | 129.1836 | 72.0366 | 124.3545 |

TABLE VI
ABLATION STUDY ON PILS

| Model | MAPE | RMSE (%) | PDE | TDE | EDP | SDP | Total error |
|---|---|---|---|---|---|---|---|
| PIL-GAN | 1.5355 | 5.2250 | 0.4683 | 8.9834 | 0.3851 | 14.6746 | 31.2719 |
| Without energy conservation | 2.8443 | 7.8811 | 1.2134 | 11.1976 | 1.8647 | 21.4532 | 46.4540 |
| Without load fluctuation | 2.0201 | 6.5387 | 0.8539 | 10.5292 | 1.0223 | 17.7854 | 38.7496 |
| Without peak load constraint | 2.2272 | 6.9510 | 0.9296 | 10.8323 | 1.2437 | 18.6957 | 40.8793 |
| GAN (no physics) | 4.1390 | 13.6435 | 2.7695 | 14.8732 | 3.4379 | 36.6592 | 75.5223 |

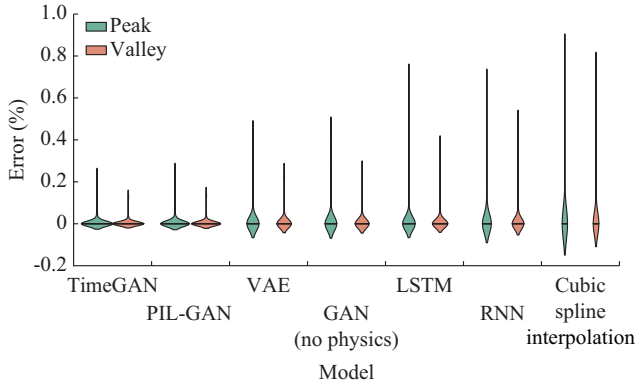


Fig. 15. Error distributions of peak and valley inpainting tasks for different models.

In addition, the PIL-GAN is compared with TimeGAN. Although PIL-GAN slightly underperforms TimeGAN, its results are still comparable, demonstrating a competitive performance in the inpainting task. The inclusion of PIL enables the model to generate more physically consistent load profiles. Furthermore, ablation studies on the PILs focus on the critical role of these physics-based constraints in improving the performance of PIL-GAN.

PIL-GAN demonstrates the potential to deal with complex scenarios. Despite the inclusion of relatively simple physics-informed components, promising results can be obtained. The physics-informed features of the PIL-GAN are not fully utilized, because the current inpainting problem lacks precisely defined physical differential equations. This suggests that incorporating the physics-informed aspect into TimeGAN can lead to the development of a new SOTA model with versatile applications.

Furthermore, the PIL-GAN can be applied to other problems such as optimal power flow analysis, in which highly complex differential equations such as the swing equation, and several other power system intricacies can be incorporated into the PIL-GAN architecture.

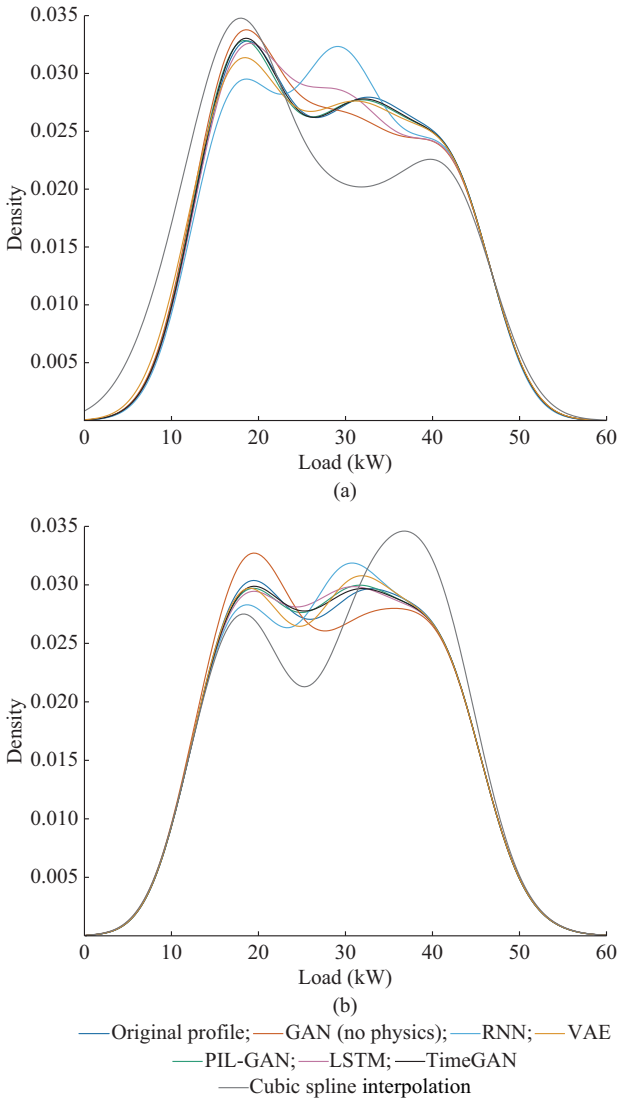


Fig. 16. Probability distributions of predicted and true values for different models. (a) Peak load value. (b) Valley load value.

Although the applicability and effectiveness of the PIL-GAN for solving real-world problems can be significantly increased by integrating sophisticated physical laws, their integration creates technical challenges.

Overall, the PIL-GAN provides a promising path for energy management systems, particularly when dealing with missing or sparse data, where it is critical to maintain physical consistency. In future studies, this model will be scaled to larger datasets and extended to incorporate additional physical laws, particularly distinct energy domains, to improve its robustness and utility for more complex applications.

## V. Conclusion

To generate high-resolution load profiles and perform the inpainting, PIL-GAN is introduced, which is a novel model that integrates GANs with physical constraints, allowing the system to learn from both the data and the underlying physical laws. The results of this paper show that the PIL-GAN outperforms traditional GAN, LSTM, RNN, VAE, and cubic spline interpolation in high-resolution load profile generation and inpainting tasks. The addition of PIL improves accuracy and physical consistency, which is confirmed in the ablation study. PIL is derived from PINN and incorporated into the loss optimization of GAN to improve the results as comparable to those of the TimeGAN. Hence, the incorporation of TimeGAN with physical constraints can yield a hybrid model with superior results.

PIL-GAN can also be applied in other areas of power systems such as optimal power flow analysis and PMU dataset generation. By integrating additional physical equations (e.g., the swing equation), PIL-GAN can comply with complex power system dynamics and generate high-resolution datasets using fewer PMUs.

Overall, the PIL-GAN offers a practical solution for high-resolution load profile generation and inpainting. It improves the accuracy of generated data while preserving physical consistency by fusing data-driven techniques with an awareness of the physical system. This bridges the gap between machine learning and practical applications by enabling more reliable and effective energy system analyses.

## References


[1] J. Wang, F. Gao, Y. Zhou *et al*., "Data sharing in energy systems," *Advances in Applied Energy*, vol. 10, p. 100132, Jun. 2023.

[2] G. Kalogridis, R. Cepeda, S. Z. Denic *et al*., "ElecPrivacy: evaluating the privacy protection of electricity management algorithms," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 750-758, Dec. 2011.

[3] T. Bailey, A. Maruyama, and D. Wallance. (2024, Oct.). The energy-sector threat: how to address cybersecurity vulnerabilities. [Online]. Available: https://www. mckinsey. com/capabilities/risk-and-resilience/our-insights/the-energy-sector-threat-how-to-address-cybersecurity-vulnerabilities

[4] J. Pei, "A survey on data pricing: from economics to data science," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4586-4608, Oct. 2022.

[5] S. Misyris, A. Venzke, and S. Chatzivasileiadis. (2020, Jan.). Physics-informed neural networks for power systems. [Online]. Available: https://arxiv.org/abs/1911.03737

[6] C. Huang, F. Li, L. Zhan *et al*., "Data quality issues for synchrophasor applications, part II: problem formulation and potential solutions," *Journal of Modern Power Systems and Clean Energy*, vol. 4, no. 3, pp. 353-361, May 2016.

[7] T. Kang, D. Y. Lim, H. Tayara *et al*., "Forecasting of power demands using deep learning," *Applied Sciences*, vol. 10, no. 20, p. 7241, Oct. 2020.

[8] W. Liao, B. Bak-Jensen, J. R. Pillai *et al*., "Data-driven missing data imputation for wind farms using context encoder," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 4, pp. 964-976, Jul. 2022.

[9] S. N. Hussain, "A novel framework based on CNN-LSTM neural network for prediction of missing values in electricity consumption time-series datasets," *Journal of Information Processing Systems*, vol. 18, no. 1, pp. 115-129, Feb. 2022.

[10] F. Ali Agga, S. A. Abbou, Y. E. Houm *et al*., "Short-term load forecasting based on CNN and LSTM deep neural networks," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 777-781, Jul. 2022.

[11] C. Bülte, M. Kleinebrahm, H. Ü. Yilmaz *et al*., "Multivariate time series imputation for energy data using neural networks," *Energy and AI*, vol. 13, p. 100239, Jul. 2023.

[12] F. Liu and C. Liang, "Short-term power load forecasting based on AC-BiLSTM model," *Energy Reports*, vol. 11, pp. 1570-1579, Jun. 2024.

[13] X. Lu, L. Yuan, R. Li *et al*., "An improved Bi-LSTM-based missing value imputation approach for pregnancy examination data," *Algorithms*, vol. 16, no. 1, p. 12, Jan. 2023.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Cambridge: MIT Press, 2014, pp. 2672-2680.

[15] I. Goodfellow. (2016, Dec.). NIPS 2016 tutorial: generative adversarial networks. [Online]. Available: https://arxiv.org/abs/1701.00160

[16] Y. Luo, X. Cai, Y. Zhang *et al.* (2018, Dec.). Multivariate time series imputation with generative adversarial networks. [Online]. Available: http://papers. nips. cc/paper/7432-multivariate-time-series-imputation-with-generative-adversarial-networks.pdf

[17] W. Wang, X. Liu, and H. He, "Synthetic load profiles for privacy-preserving energy data sharing using GANS," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2331-2342, May 2021.

[18] A. Pinceti, L. Sankar, and O. Kosut, "Synthetic time-series load data via conditional generative adversarial networks," in *Proceedings of 2021 IEEE PES General Meeting*, Washington DC, USA, Jul. 2021, pp. 1-5.

[19] Y. Hu, Y. Li, L. Song *et al*., "MultiLoad-GAN: a GAN-based synthetic load group generation method considering spatial-temporal correlations," *IEEE Transactions on Smart Grid*, vol. 15, no. 2, pp. 2309-2320, Mar. 2024.

[20] A. Madhubalan, A. Gautam, and P. Tiwary, "Blender-GAN: multi-target conditional generative adversarial network for novel class synthetic data generation," in *Proceedings of 2024 International Conference on Smart Applications*, *Communications and Networking*, Harrisonburg, USA, May 2024, pp. 1-7.

[21] D. Upadhyay, Q. Luo, J. Manero *et al*., "Comparative analysis of tabular generative adversarial network (GAN) models for generation and validation of power grid synthetic datasets," in *Proceedings of IEEE 20th International Conference on Smart Technologies*, Torino, Italy, Jul. 2023, pp. 677-682.

[22] V. Avkhimenia, T. Weis, and P. Musilek, "Generation of synthetic ampacity and electricity pool prices using generative adversarial networks," in *Proceedings of 2021 IEEE Electrical Power and Energy Conference*, Toronto, Canada, Oct. 2021, pp. 225-230.

[23] X. Zheng, B. Wang, and L. Xie, "Synthetic dynamic PMU data generation: a generative adversarial network approach," in *Proceedings of 2019 International Conference on Smart Grid Synchronized Measurements and Analytics*, College Station, USA, May 2019, pp. 1-6.

[24] L. Song, Y. Li, and N. Lu, "ProfileSR-GAN: a GAN based super-resolution method for generating high-resolution load profiles," *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 3278-3289, Jul. 2022.

[25] N. Ahmed and L. Schmidt-Thieme, "Sparse self-attention guided generative adversarial networks for time-series generation," in *Proceedings of 2023 IEEE 10th International Conference on Data Science and Advanced Analytics*, Thessaloniki, Greece, Oct. 2023, pp. 1-2.

[26] J. Lan, Y. Zhou, Q. Guo *et al*., "A data-driven approach for generating load profiles based on InfoGAN and MKDE," *Frontiers in Energy Research*, vol. 11, p. 1339543, Dec. 2023.

[27] R. Hachache, M. Labrahmi, A. Grilo *et al*., "Energy load forecasting techniques in smart grids: a cross-country comparative analysis," *Energies*, vol. 17, no. 10, p. 2251, May 2024.

[28] Y. Li, L. Song, Y. Hu *et al*., "Load profile inpainting for missing load data restoration and baseline estimation," *IEEE Transactions on Smart Grid*, vol. 15, no. 2, pp. 2251-2260, Mar. 2024.

[29] J. Wang, W. Du, W. Cao *et al*. (2025, May). Deep learning for multivariate time series imputation: a survey. [Online]. Available: https://arxiv.org/abs/2402.04059
[30] J. Yoon, D. Jarrett, and M. van der Schaar. (2019, Dec.). Time-series generative adversarial networks. [Online]. Available: https://www.damtp.cam.ac.uk/user/dkj25/pdf/yoon2019time.pdf
[31] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686-707, Feb. 2019.
[32] S. Zhao, Y. Peng, Y. Zhang *et al*., "Physics-informed machine learning for parameter estimation of DC-DC converter," in *Proceedings of 2022 IEEE Applied Power Electronics Conference and Exposition*, Houston, USA, Mar. 2022, pp. 324-329.
[33] Y. Fassi, V. Heiries, J. Boutet *et al*., "Toward physics-informed machine-learning-based predictive maintenance for power converters: a review," *IEEE Transactions on Power Electronics*, vol. 39, no. 2, pp. 2692-2720, Feb. 2024.
[34] Pecan Street Inc. (2018, Dec.). Dataport 2018. [Online]. Available: https://www.pecanstreet.org/dataport/

**Swodesh Sharma** received the B.Eng. degree in electrical engineering from Tribhuvan University, Institute of Engineering, Pulchowk, Nepal, in 2024, and is currently pursuing the Ph.D. degree in electrical and computer engineering at Florida International University, Miami, USA. His research interests include cyber-physical system, electromagnetic transient simulation, and application of artificial intelligence in power systems.

**Apeksha Ghimire** received the B.Eng. degree in electrical engineering from Tribhuvan University, Institute of Engineering, Pulchowk, Nepal, in 2024. She is working as an Associate Electrical Engineer in the Power System Studies and Planning Department at K&A Engineering Consulting, P.C., a U.S.-based company, where she serves at its Nepal branch in Balkhu, Kathmandu, Nepal. Her research interests include renewable energy, grid integration, power system operation, and energy policy for system reliability and sustainability.

**Shashwot Shrestha** received the B. Eng. degree of electrical engineering from Tribhuvan University, Institute of Engineering, Pulchowk, Nepal, in 2024. His research interests include power system, power electronics, smart grid, microgrid, power system optimization, cybersecurity, artificial intelligence, and electric vehicle.

**Rachana Subedi** received the Bachelor's degree in electrical engineering from Tribhuvan University, Institute of Engineering, Pulchowk, Nepal, in 2024. Her research interests include power system, grid integration of renewable energy, power electronic converter, power system protection, electric vehicle, and artificial intelligence.

**Sushil Phuyal** received the B.E. degree in electrical engineering from Tribhuvan University, Institute of Engineering, Pulchowk, Nepal, in 2024. His research interests include converter topology, resonant converter, optimization and control, power circuit miniaturization, and electronic circuit design.