2023 MagNet Challenge Webinar: Machine Learning Methods

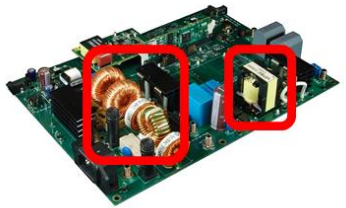
Haoran Li
Princeton University

May. 19, 2023

Magnetic Components

- Power magnetics are critical in power electronics system.
- Typically occupy the largest volume and introduce significant power loss, limiting the system optimization.

LED Drivers

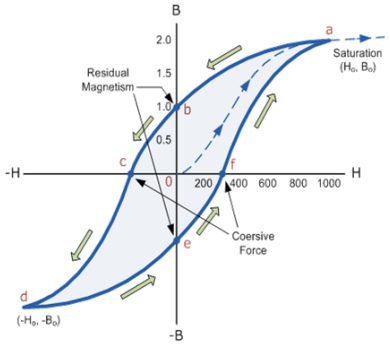


Solar Inverters

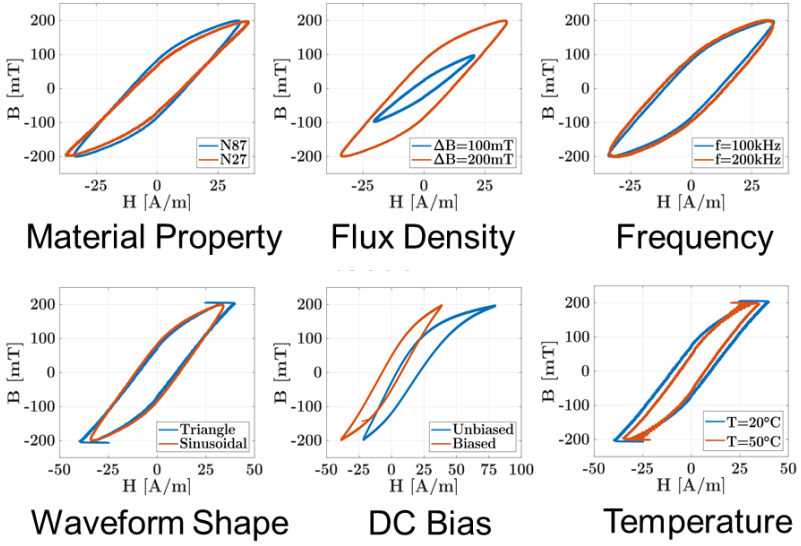
Voltage Regulator for GPUs

Challenges in Magnetics Modeling

- Modeling the behaviors of power magnetics, especially the hysteresis loop and the core loss, is challenging.
- The behavior of power magnetics can be impacted by various factors, e.g., waveform, frequency, temperature and dc bias.



Area of B-H loop → Loss



Methods for Core Loss Modeling

- Steinmetz Equation *three parameters, sine wave*

$$P_v = k f^{\alpha} \hat{B}^{\beta}$$

Hard to cover all the influencing factors

- Improved GSE (iGSE) *three parameters, arbitrary wave*

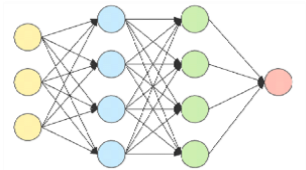
$$P_v = \frac{1}{T}\int_0^T k_i \left|\frac{dB}{dt}\right|^{\alpha} (\Delta B)^{\beta-\alpha}\, dt$$

- Improved – improved GSE (i2GSE) *eight parameters*

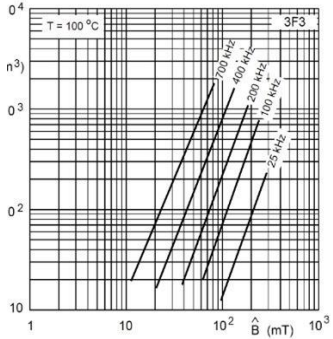
$$P_v = \frac{1}{T}\int_0^T k_i \left|\frac{dB}{dt}\right|^{\alpha} (\Delta B)^{\beta-\alpha}\, dt + \sum_{l=1}^{n} Q_{rl} P_{rl}$$

- Neural networks *thousands of parameters*

DC Bias, Temperature, Waveform



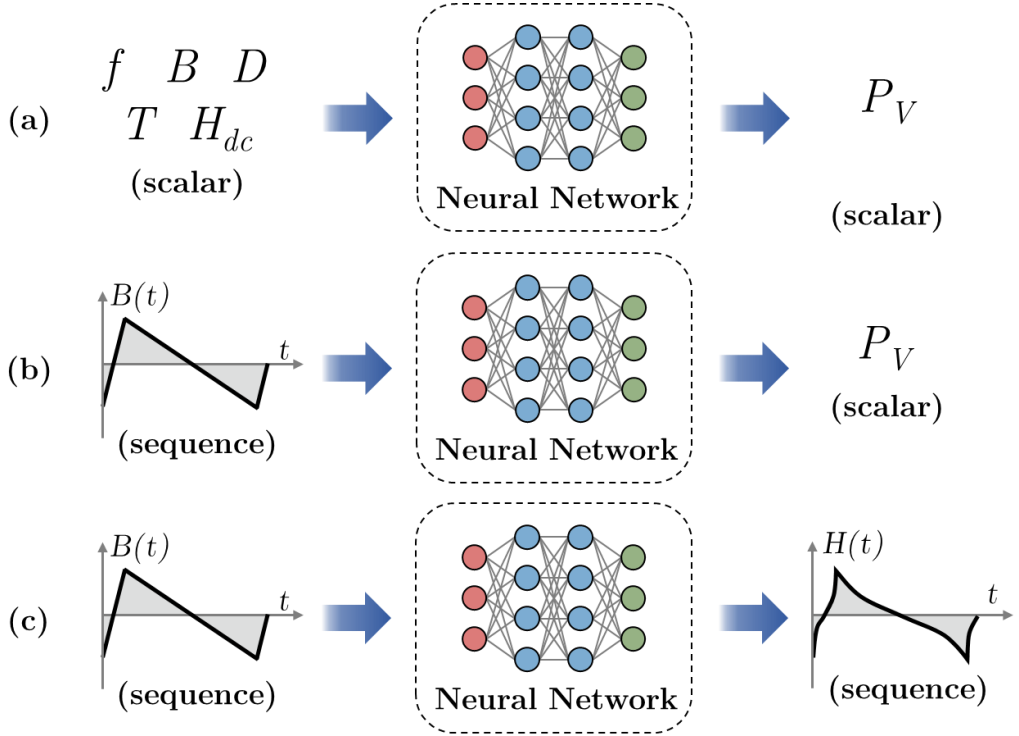
Neural Networks



Core Loss

Neural Network Models

- Different problem settings → different inputs/outputs → different network structure



- The goal of these examples is to highlight the typical workflow of neural network modeling for power magnetics, not meant to be comprehensive or accurate.
- The dataset and algorithm in the tutorial is not completely the same as the one for the challenge.
- Synthesized based on PyTorch framework.
- Tutorial notebooks are available in GitHub: https://github.com/PrincetonUniversity/magnet/tree/main/tutorial

Workflow of Network Training

PRINCETON UNIVERSITY

- Step-1: pre-process the dataset as needed.

- Step-2: randomly split the dataset into training set, validation set, and test set.

  - K-fold validation is recommended to improve the generalization capability

- Step-3: define the neural network structure.

  - Pack the network model into a standard function with required inputs/outputs

- Step-4: define the training iteration.

  - Wisely select the loss function, the learning rate and its scheduler, and the optimizer

- Step-5: train the neural network → evaluate the results → iterate the training → fine-tune

Pre-processing of Dataset

PRINCETON UNIVERSITY

- Tags for waveform shapes are not included in the provided dataset. Teams are expected to develop models for arbitrary waveforms (sinusoidal, triangular, trapezoidal for now), or develop their own classification algorithm if needed.

- Different fields of data (B, f, T, H, P_V) have significantly different magnitudes. Data normalization and transformation (e.g., $P_V \rightarrow \log(P_V)$) are typically preferable.

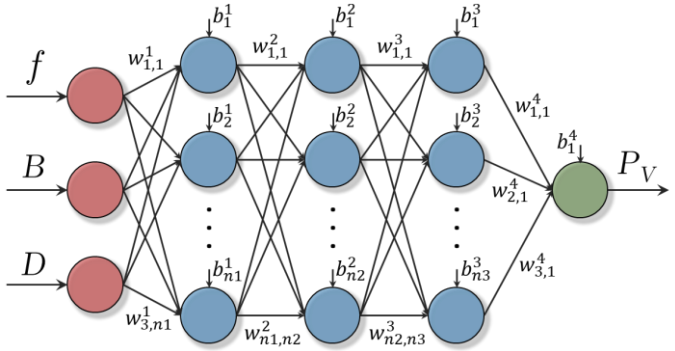
- Waveforms (B and H) are provided in single-cycle format. Assigning random phases and reasonable white noise will help to improve the generalization capability.

- Some of the developed pre-processing tools in MATLAB are provided in GitHub for reference: https://github.com/PrincetonUniversity/magnet/tree/main/tutorial

Example-1: Feedforward Network

- Scalar inputs: frequency, flux density amplitude, temperature, duty ratio, etc.
- Scalar outputs: predicted core loss.
- Demo: predicting the core loss for triangular waves with different duty ratios.

$$P_v = k f^{\alpha} \hat{B}^{\beta}$$

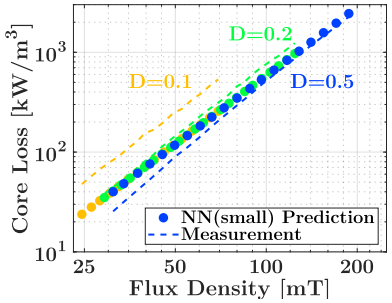


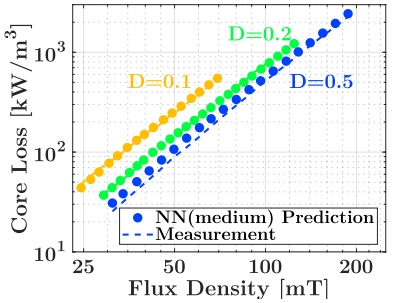
- Pro: straightforward to implement, simple neural network structure, easy to train.
- Con: losing the information of the waveform shapes.

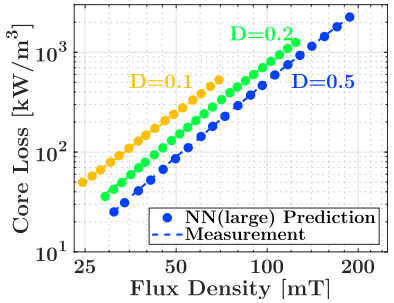
Example-1: Feedforward Network

PRINCETON UNIVERSITY

- Evaluation of the network performance in terms of the core loss prediction.







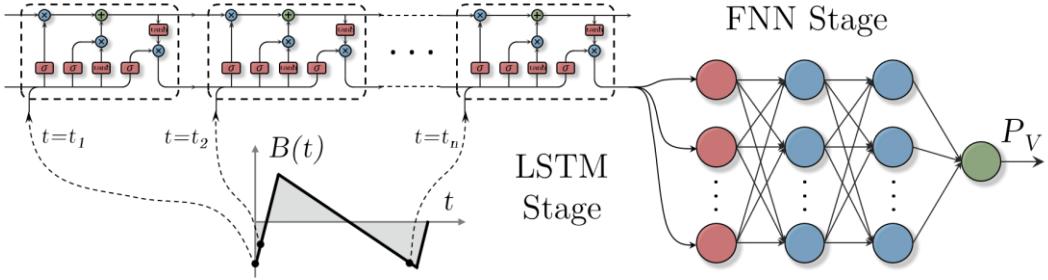
Neurons in the Each Layer			Total Number of Parameters	Abs. Avg. of Relative Error
2	1	3	21	19.76%
5	8	4	109	9.81%
15	15	9	454	5.33%
29	27	23	1594	1.81%
44	57	47	5515	1.77%

- For different duty ratios of triangular waveforms, the core loss is predicted with reasonable accuracy. Network performance is closely related to the network size.

Example-2: LSTM Network

- Sequence inputs: waveform of excitation (flux density $B(t)$ in this case).
- Scalar outputs: predicted core loss.
- Demo: predicting the core loss for mixed waves.

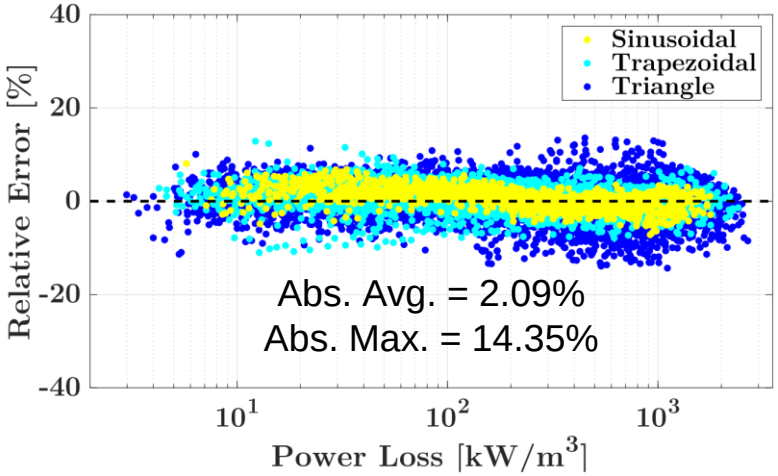
$$P_v = \frac{1}{T}\int_0^T k_i \left|\frac{\mathrm{d}B}{\mathrm{d}t}\right|^{\alpha} (\Delta B)^{\beta-\alpha}\,\mathrm{d}t$$



- Pro: naturally taking the waveform shape into consideration.
- Con: potentially difficult to incorporate other conditions, such as the temperature.

Example-2: LSTM Network

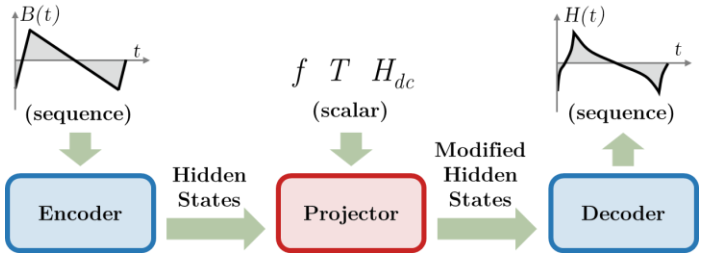
- Evaluation of the network performance in terms of the core loss prediction.



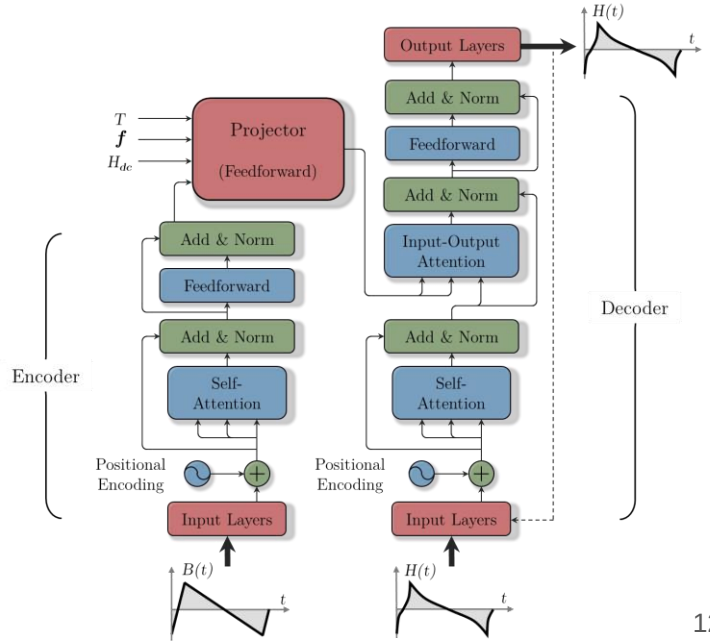
- For all the three types of waveform shapes, the core loss is predicted with reasonable accuracy using a single neural network model.

Example-3: Transformer Network

- Sequence inputs: waveform of excitation (flux density $B(t)$ in this case).
- Sequence outputs: waveform of response (field strength $H(t)$ in this case).
- Demo: predicting the B-H loop under various operating conditions.



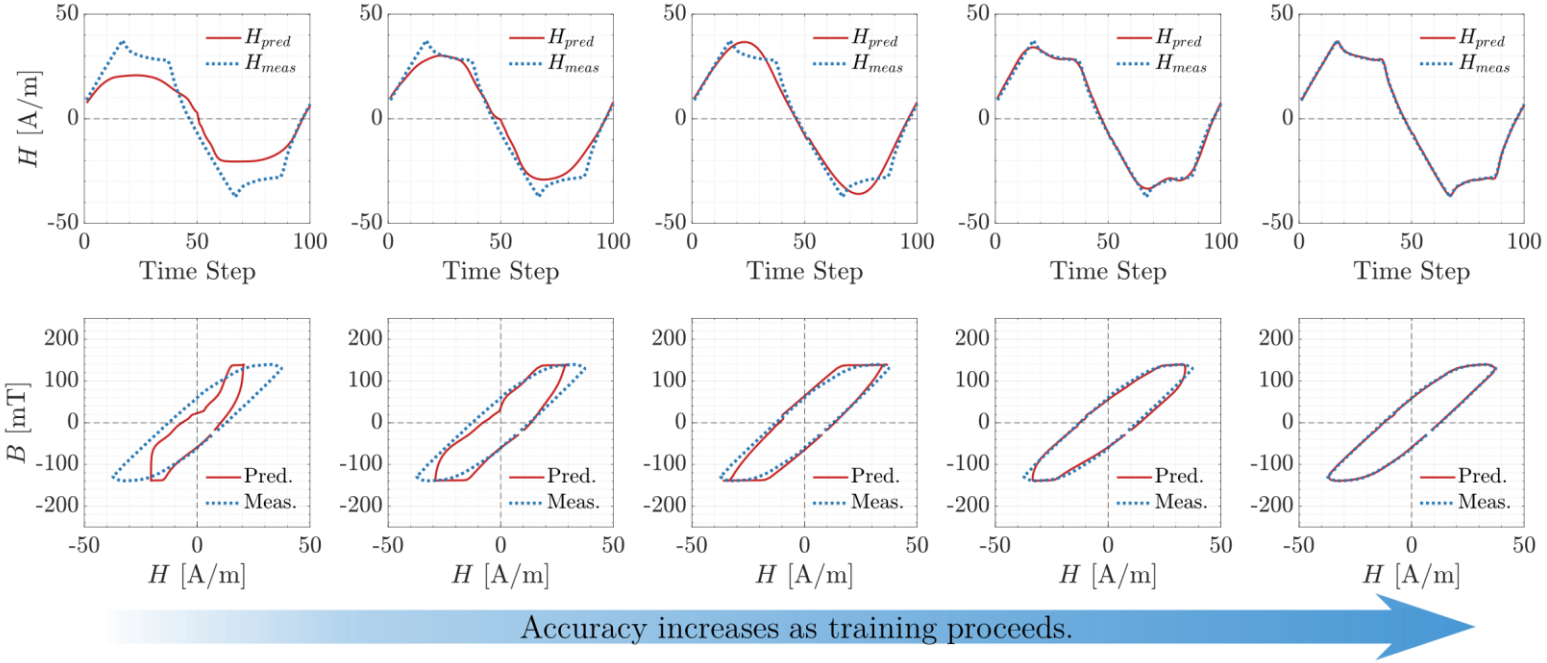
- Pro: combine the sequence inputs and the scalar inputs into an integrated framework.
- Con: the information of core loss is not directly taken into consideration



Example-3: Transformer Network

PRINCETON UNIVERSITY

- As the training proceeds, the model gradually converges, eventually providing an accurate prediction of the B-H loop.



Example-3: Transformer Network

PRINCETON UNIVERSITY

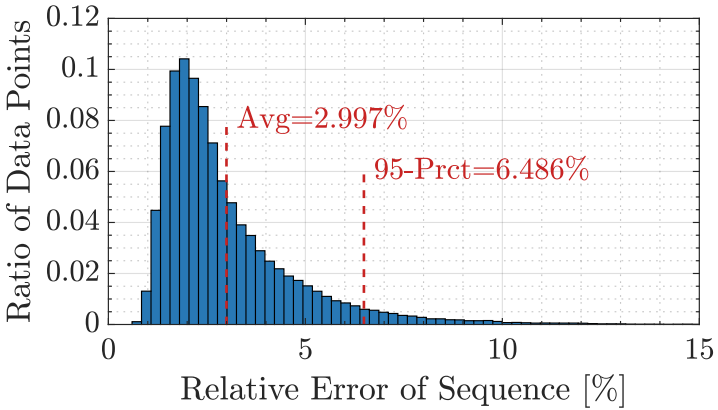
- Evaluation of the network performance in terms of the sequence prediction and the core loss prediction.

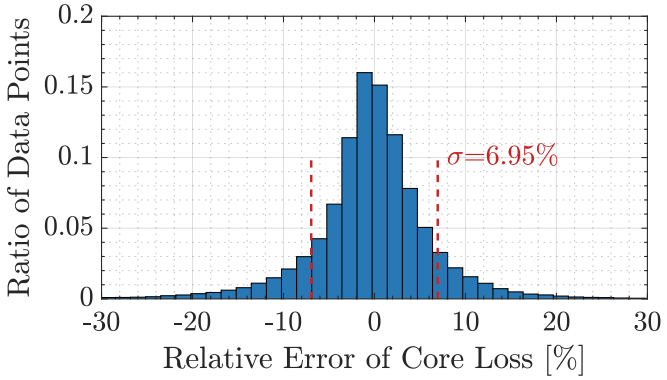
$$\text{Relative Err. of Sequence} = \frac{\text{rms}(H_{pred} - H_{meas})}{\text{rms}(H_{meas})}$$

$$= \frac{\sqrt{\frac{1}{n}\sum_{t=t_1}^{t_n}\left(H_{pred}(t) - H_{meas}(t)\right)^2}}{\sqrt{\frac{1}{n}\sum_{t=t_1}^{t_n}\left(H_{meas}(t)\right)^2}}$$

$$P_V = \frac{1}{T}\int_{B(0)}^{B(T)} H(t)\, dB(t)$$

$$\text{Relative Err. of Core Loss} = \frac{|P_{V,pred} - P_{V,meas}|}{P_{V,meas}}$$





Tips for Network Training

PRINCETON UNIVERSITY

- Properly balance the trade-off between the model size and the model performance. Optimize the hyperparameters using existing tools such Optuna.

- Different magnetic materials have different behaviors, which potentially leads to different difficulties to model. Comprehensively train and test on the 10 available material datasets and fine-tune for the testing materials.

- Be aware of the overfitting during the training and ensure the generalization capability. Keep an eye on the metrics and the distribution of prediction errors.

- Tricks of setting the loss function and learning rates may significantly affect the training results and network performance.

PRINCETON UNIVERSITY

Thank you for your interest!

IEEE Advancing Technology for Humanity

PRINCETON UNIVERSITY

pels IEEE POWER ELECTRONICS SOCIETY Powering a Sustainable Future

Dartmouth

https://mag-net.princeton.edu/

https://github.com/PrincetonUniversity/magnet

https://github.com/minjiechen/magnetchallenge