

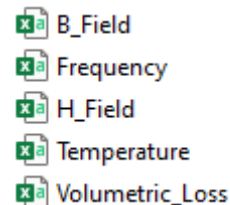
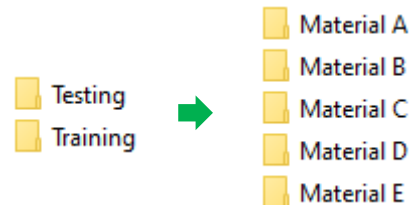
MagNet Challenge Final Evaluation Office Hour

Nov. 30, 2023

- November 10th, 2023 – Data released for final evaluation.
 - 1) Download the new training data and testing data from the following link for the 5 additional materials:
https://www.dropbox.com/sh/q5w2ddol8y6bk0k/AABXKxv_aiLj8yXspeusJq4na?dl=0
 - 2) Train, tune, and refine your model or algorithm using the training data.
 - 3) Predict the core losses for all the data points contained in the testing data for the 5 materials. For each material, the prediction results should be formatted into a CSV file with a single column of core loss values. Please make sure the index of these values is consistent with the testing data, so that the evaluation can be conducted correctly.
- December 31st, 2023 – Final submission.
 - 1) Prediction results for the testing data are due as 5 separate CSV files for the 5 materials.
 - 2) A 5-page IEEE TPEL format document due as a PDF file. Please briefly explain the key concepts.
 - 3) Full executable model due as a ZIP file for a potential code review with winning teams.
 - 4) Submit all the above required files to pelsmagnet@gmail.com.
- January to March 2024 – Model Performance Evaluation, Code Review, Final Winner Selection

Final Evaluation Dataset

- 5 new materials are provided, which are kept anonymous and labeled as A, B, C, D, and E.
- Training data contains 5 CSV files, including: **B_Field** ($N \times 1024$, in T), **H_Field** ($N \times 1024$, in A/m), **Frequency** ($N \times 1$, in Hz), **Temperature** ($N \times 1$, in $^{\circ}\text{C}$), and **Volumetric_Loss** ($N \times 1$, in W/m^3). N refers to the number of data points.
- Testing data contains 3 CSV files, including: **B_Field** ($N \times 1024$, in T), **Frequency** ($N \times 1$, in Hz), and **Temperature** ($N \times 1$, in $^{\circ}\text{C}$). Similarly, N refers to the number of data points.



Final Evaluation Dataset

- The number of data points for each material is provided in the following table:

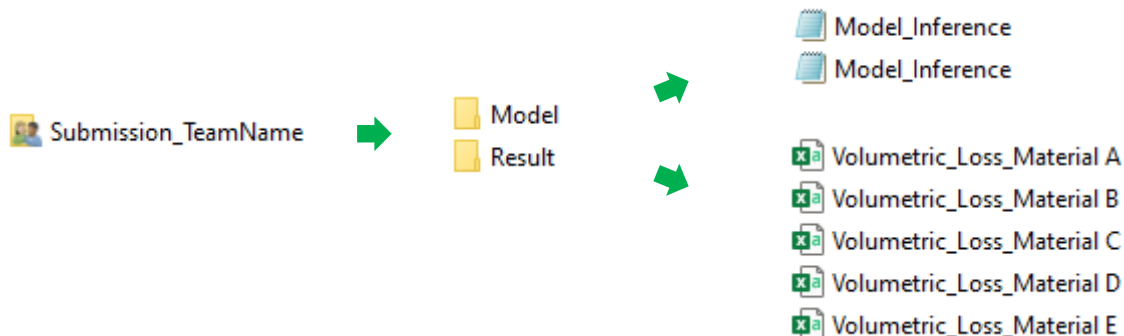
Number of Data Points	Material A	Material B	Material C	Material D	Material E
Training Data	2432 (101/694/1637)	7400 (364/2253/4783)	5357 (215/1679/3463)	580 (145/400/35)	2013 (57/667/1289)
Testing Data	7651 (334/2174/5143)	3172 (147/980/2045)	5357 (212/1751/3394)	7299 (61/2247/4991)	3738 (107/1205/2426)

(The numbers in brackets denote the number of data points for sinusoidal waves, triangular waves, and trapezoidal waves, respectively.)

- Please note that the training data and testing data are **intentionally split and selected in specific ways**, different from material to material, for the purpose of the MagNet challenge. **Unbalanced data distribution** is normal and should be considered when developing / refining your models.
- The complete dataset will be released after the MagNet challenge is completed.

Result Submission

- Please follow the steps below to ensure a smooth and correct evaluation of your results:
 - Rename the folder name with your team's name (e.g., the name of the university).
 - Put all your codes and related files into the subfolder "Model".
 - Put all the output files into the subfolder "Result".



- For Python-based codes, follow the template in "Model_Inference.py". Or for MATLAB-based codes, follow the template in "Model_Inference.m".
 - Wrap your model or algorithm into the function "core_loss", with inputs of **data_B** (flux density waveform, N by 1024, in T), **data_F** (frequency, N by 1, in Hz), and **data_T** (temperature, N by 1, in C).
 - In the function "core_loss", the volumetric core loss values will be calculated by your model or algorithm, and should be **formatted in the shape of N by 1 with the unit of W/m³**. The results are then written into a csv file. Please make sure the **index order in the output file is consistent** with that of the inputs. To guarantee a proper code evaluation, please also make sure the output file generated by this function is **the same as the output file** you submit.
 - For any functions called from other files, please use the **relative path address** to indicate the file locations, and make sure these functions can be correctly loaded and compiled.
 - Python users: for any functions called from **other modules or packages**, please create a **requirement.txt** file to indicate their **names and versions**. Please make sure these modules and packages are **open sourced and accessible**.
 - MATLAB users: for any functions called from any **other specific toolboxes**, please create a **requirement.txt** file to indicate their **names and versions**. Please make sure these toolboxes are **public and accessible**.

```

1 import random
2 import numpy as np
3
4 Material = "Material A"
5
6 %%% Load Dataset
7
8 def load_dataset(in_file1='./Testing/'+Material+'/B_Field.csv',
9                 in_file2='./Testing/'+Material+'/Frequency.csv',
10                 in_file3='./Testing/'+Material+'/Temperature.csv'):
11
12     data_B = np.genfromtxt(in_file1, delimiter=',') # N by 1024, in T
13     data_F = np.genfromtxt(in_file2, delimiter=',') # N by 1, in Hz
14     data_T = np.genfromtxt(in_file3, delimiter=',') # N by 1, in C
15
16     return data_B, data_F, data_T
17
18 %%% Calculate Core Loss
19 def core_loss(data_B, data_F, data_T):
20
21     #===== Wrap your model or algorithm here=====
22
23     # Here's just an example:
24
25     length = len(data_F)
26     data_P = np.random.rand(length,1)
27
28     #=====
29
30     with open('./Testing/Volumetric_Loss_'+Material+'.csv', "w") as file:
31         np.savetxt(file, data_P)
32         file.close()
33
34     print('Model inference is finished!')
35
36     return
37
38 %%% Main Function for Model Inference
39
40 def main():
41
42     # Reproducibility
43     random.seed(1)
44     np.random.seed(1)
45
46     data_B, data_F, data_T = load_dataset()
47
48     core_loss(data_B, data_F, data_T)
49
50 if __name__ == "__main__":
51     main()
52
53 # End

```

```

1 clear all; close all; clc;
2
3 rng(1); % Reproducibility
4
5 global Material
6 Material = 'Material A';
7
8 main();
9
10 %%% Run Main
11 function main()
12
13     [data_B, data_F, data_T] = load_dataset();
14     core_loss(data_B, data_F, data_T);
15
16 end
17
18 %%% Load Dataset
19 function [data_B, data_F, data_T] = load_dataset()
20
21     global Material
22
23     data_B = readmatrix ['./Testing/', Material, '/B_Field.csv']; % N by 1024, in T
24     data_F = readmatrix ['./Testing/', Material, '/Frequency.csv']; % N by 1, in Hz
25     data_T = readmatrix ['./Testing/', Material, '/Temperature.csv']; % N by 1, in C
26
27 end
28
29 %%% Calculate Core Loss
30 function core_loss(data_B, data_F, data_T)
31
32     global Material
33
34     #===== Wrap your model or algorithm here=====
35
36     % Here's just an example:
37
38     len = length(data_F);
39     data_P = rand(len,1);
40
41     #=====
42
43     writematrix(data_P, ['./Testing/Volumetric_Loss_', Material, '.csv']);
44     fprintf('Model inference is finished! \n');
45
46 end
47
48
49 %%% End

```

- For codes in languages other than Python or MATLAB, please try to pack your codes in a similar way as above. Make sure that the codes are **fully executable**, and can generate **the same output files** as those you submit.

- Put the report (.pdf file) into the folder. Please make sure to report the total number of model parameters, indicating your model size as a table in the document. These numbers will be confirmed during the code review process.
 - In this Challenge, we consider all the values that are related to the magnetic material characteristics as the parameters of the model, such as all the learnable parameters in the neural networks, or the fitting coefficients in the analytical models. Others such as those related to the indexing are not considered.
 - In your final report, please include the numbers of parameters in your model for each of the materials (A, B, C, D, and E). The table below shows an example:

Material	A	B	C	D	E
Number of Parameters	16295	18116	18268	12648	19151

- In the end, pack the entire folder as a ZIP file for submission.

Any questions?

Frequently asked questions will be recorded and documented in the MagNet Challenge GitHub repo for your future reference.