MagNet Challenge Final Submission Report

Emmanuel Havugimana ehavugim@asu.edu
Vivek Thomas Chacko vtchacko@asu.edu
Sritharini Radhakrishnan sradha14@asu.edu
Jorge Peralta japeral2@asu.edu
Mike Ranjram Mike.Ranjram@asu.edu

***Abstract*—Participating teams of the 2023 MagNet Challenge compete to create a software package that utilizes the magnetic characteristics of different materials to predict their core loss efficiently and with high accuracy. The difficulty of this challenge is in using the developed model to accurately predict the core loss of new materials that it has not previously trained on. Over the course of the past few months, the ASUMag team has carried out exploratory analysis of the core loss computation equations and applied data-driven machine learning (ML) algorithms to develop a novel core loss-prediction model. Preliminary results were shared in our Nov. 1st submission. In our final submission, we refine our model to reduce its size and execution time significantly. Our main aim in this challenge is to produce a model of reasonable accuracy while minimizing its size. We believe we have successfully achieved this goal through the model design choices described below. In the final data set, we observed a subpar performance of our model on new material, noticeably a logarithmic relationship between error and data size. We attribute this observation to the size difference in the provided training sets from different materials than material differences. The data provided to train the model was significantly smaller than the data provided to the pre-test phase, which we assume to be the root cause of our model's performance, especially since any error in our model in training is only amplified in testing due to this data size imbalance. Our model's performance will revert to its nominally accurate core-loss predictions with a larger or more representative training data set.**

***Index Terms*—core loss, MagNet challenge, Logarithm, re-sample, Feed Forward Neural Network**

I. Introduction

In this paper, a model is proposed and compared with baseline models from MagNet AI [1] and equation-based models from MagNet-hosted tutorials [5]. Specifically, the ASUMag model is compared with accuracy limits stipulated by the 2023 MagNet Challenge [6]. The results are promising. Our models for each material share the same architecture, only differing in weight. Our material-independent architecture results from the inherent similarity of nominal core loss prediction, the physical basis of our loss-informed model architecture. Such modeling allows for cross-material learning as material extension becomes easier as one model is optimized. This allows for material similarity analysis and model versioning, which only involves updating the weights. These model weights are shared as JSON files for human readability and cross-platform support, making them easy to use. The model shared in the report is represented in a form that makes it easy to transfer across platforms.

II. Literature review

The Princeton team reported errors based on the B-H loop and core loss [1]. The model results are compared with models from Princeton team papers mainly [1]. The 95th percentile is considered equivalent to 2σ where standard deviation σ was reported instead of the 95th percentile and used half-normal distribution [4] approximations for deriving metrics from other metrics. This allows for the comparison of results from the literature and MagNet competition results. Even in cases where a subset of MagNet competition metrics are reported, a proxy can be computed for other metrics.

The results indicate a general model with performance that scales with training data. The estimates of the 95th percentile are from comparing the prediction of the model to the provided training set.

III. Model description

A description of ASUMag's current model is given in this section. The model is based on 2023 MagNet-held tutorials. A unique contribution of introducing down-sampling in the data processing step allows for the reduction of model size while incorporating arbitrary waveshape modeling using the Feed Forward Neural network architecture. The limits of down-sampling was set by studying the reconstruction accuracy observed and our target model performance.

Our design makes our model more independent of the 'B' wave shape and allows it to be fitted with all 'B' wave shapes or selectively by one wave shape. As we have trained the model with all wave shapes, one model is shared for a material.

1) Input to model (29x1)
    a) First 24 points sample of "B" (24x1)
    b) Temperature (4 temperature) one-hot encoding (4x1)
    c) Frequency (1x1)
2) Model: Feed Forward Neural Network with
    a) Layer 1 : 29x29, Layer 2: 29x15, Layer 3: 15x15, Layer 4: 15x1
3) Output: 1x1 core loss

IV. Model Size and Performance Estimates

The setup is a pipeline for reading data from the folders and evaluating a model with both RMS of absolute percent error and 95th percentile absolute percent error.

The 95th percentile error is computed for a model we developed for ten different materials.

this might be a result of material characteristics, data set size, waveform type, or composition ratios. Based on our exploration of model performance variation among the ten training materials since our Nov. 1st submission, we found

that it is redundant to develop different models for different materials.

We note that if there is any variation in error between the different materials for the same model, this might be a result of material characteristics, data set size, waveform type, or composition ratios. Table I shows our estimated performance per wave shape type and material. Based on our exploration of model performance variation among the ten training materials since our Nov. 1st submission, we found that it is redundant to develop different models for different materials.

Material	**A**	**B**	**C**	**D**	**E**
Model size	1576	1576	1576	1576	1576
Training data size	2432	7400	5357	580	2013
Estimate 95th perc. Err.	7.9%	5.3%	8.2%	19.3%	10.2%
Sine 95th perc. Err.	18%	8%	20%	26%	21%
Tri 95th perc. Err.	6%	5%	7%	17%	10%
Trap 95th perc. Err.	6%	5%	7%	28%	9%

TABLE I
MODEL SIZE FOR EACH MATERIAL AND ESTIMATES OF 95TH BY WAVE TYPE

V. ASUMAG MODEL CHOICES

A. Logarithmic transformations

The training loss of a model is usually reported as an absolute measure and is not relative to the size of the training data set. Therefore, we have chosen to apply a logarithmic transformation to the output of the model so that the resulting error is independent of the core loss magnitude.

This choice enables our model to meet the competition requirements of reporting percentage error without writing an additional custom loss function. Moreover, as Steinmetz's equation includes multiplicative and exponential parameters, a logarithmic transformation is suitable for model convergence besides the direct mapping of error in training and the percentage error. Although these transformations may limit model learning in the case of additive 'B' losses, we don't expect these effects to dominate in the test cases.

As a result of the logarithmic transformations we apply, we expect most of the errors during training to be less than one, which can be used to shape a histogram of errors by the custom error criterion. The observed distribution of error that seems to be size-dependent is likely due to possible data collection-related errors which inversely correlate to the measured size. As stated above, the log transform makes the percentage error size-independent; therefore, any residual 'dependence' still observed may result from the collected data's unpredictability. To rectify such anomalies requires a minor but more complex fitting trend.

From the logarithm transform,

$$value = log(Power)$$

So, an error criterion (mean) during training becomes a percentage metric in the final prediction after anti-log

$$Power\_meas = 10^{value}$$

$$Power\_pred = 10^{value+criterionLoss}$$

$$Error = |\frac{Power\_pred}{Power\_real} - 1|$$

$$Error = |10^{0+criterionLoss} - 1|$$

from $d\frac{d(10^x)}{dx} = ln(10) * 10^x$, and from expansion of $10^{0+error}$ to first order term, we get

$$Error \approx |1 + ln(10) \times criterionLoss - 1|$$

$$Error \approx ln(10) \times criterionLoss$$

This direct link between percentage error and criterion loss can be exploited in training to shape the loss histogram. We think it can be done but have not implemented this in our model.

B. The 95th percentile error highly correlates with mean error

As reported in pre-test estimates, we found a consistent scaling factor of the 95th percentile error with respect to the mean error. Hence, during optimization, we did not create a new training loss for the 95th percentile metric. An error distribution can be skewed to be more 95th-percentile friendly, but not by much. Assuming a normal distribution of errors and a zero average mean (bias), the absolute error distribution gives a guide to transforming error reports missing the 95th percentile for comparison with the competition metric.

C. Simplified model structure

Our goal in the competition is to minimize the size of our model while still achieving reasonable results in accuracy. To achieve this goal, we have chosen to track only a few key parameters as dictated by the model equation per new material. We argue that one must also consider model storage when considering model size. Hence, one can save on model storage space by reducing the number of parameters to store when training a new material.

1) Same data processor for all materials: Material-dependent data transformations make data more palatable to train the model, but one must store normalization factors to remember the original form of the data, thus increasing the model size. Without such transformations, computation times do not necessarily get longer; however, we may sacrifice some model accuracy because we don't normalize the data.

All our model information is in the model weights; to add a new material, only the model weights must be added.

2) Transfer learning: With the same model structure, the material cross-learning is improved through initializing weights with other material models via material similarity analysis as in table III.

In table III, the 95th percentile is highly dependent on the trained material and variance across model initialization. The small error, when initialized with a material, might indicate some similarity between the two materials. This shows that with a growing repository of materials, new materials would easily be found by finding similar materials to initialize weights or do other training techniques with.

TABLE II

New Material /Old material	3C90	3C94	3.E6	3F4	77	78	N27	N30	N49	N87	Min	Max
A	13%	12%	11%	17%	8%	12%	13%	11%	12%	10%	8%	17%
B	6%	4%	7%	5%	7%	5%	5%	5%	5%	6%	4%	7%
C	9%	8%	8%	8%	7%	8%	9%	7%	9%	8%	7%	9%
D	27%	19%	21%	21%	22%	19%	23%	26%	32%	21%	19%	32%
E	18%	17%	14%	17%	14%	13%	15%	17%	17%	17%	13%	18%
Average	15%	12%	12%	14%	12%	11%	13%	13%	15%	12%	10%	17%

TABLE III
MODEL 95TH PERCENTILE GIVEN INITIALIZATION WITH EACH OF TEN MATERIALS

D. Model accuracy and data pre-processing error

1024 to 24 B_field transform: Our data processing stage involves down sampling 1024 data points to 24; this introduces a 2% 95th percentile error in the reconstruction of the 'B' wave shape. This was measured by reconstructing the original 1024 data points from the down-sampled 24 and comparing the differences as

$$ReconstrError = \frac{sum(abs(Breconstr - Boriginal))}{sum(abs(Boriginal))}$$

B^{β}, the error introduced, is approximately $2\% \times \beta$ approx 4% to 6%. Note: this can be reduced by using shifts of original data and re-sampling, sending all sampled shifts to the model, and looking at the results. This limitation can be traded with model size; we chose to reduce input size to reduce model size, which was acceptable given that the measurement accuracy are negligible. Our current model structure allows a designer to adapt it to be more accurate by increasing the size while keeping the architecture the same. The transformation is general in the sense it does not rely on concepts of duty cycle, zero voltage times, etc., which would assume a voltage-driven system (defined wave shape).

E. Treatment of temperature in the model

Currently, the model uses four inputs for temperature. For now, it uses one-hot encoding to encode 25, 40, 70, and 90 (the standard temperatures in the model). As a result, the model is limited by temperature in training. During testing, one can post-fit arbitrary temperatures by finding the losses at all temperatures and fitting the temperature afterward. For example, core loss at 30 degrees can be estimated by calculating loss at 25, 50, 70, and 90 degrees and then using interpolation to find the core loss at the specific temperature. We have not encountered any temperature values outside of what we have encoded during the training and testing. The one-hot temperature encoding was removed to incorporate arbitrary temperature training in the model. But given how temperature is weighed in the model as compared to 'B' or frequency, we think it might not be as useful to know core loss accuracy up to a 1-degree resolution. However, we believe the temperature encoding, as related to our model, can be improved in the future.

F. Model limitations and guidelines

This section states how not to use the model and describes potential pitfalls in the model to prevent misusage or misattribution of erroneous results.

Our model was trained on the MagNet dataset; hence, any limitations that the data might have in the loss's predictability will propagate to our model's results. The model structure should be trained and tested on the same data structures. We caution the user to use the same parameter units as the model as our current pre-processing stages do not have a unit detection system.

In the data set, the data provided was not phase-shifted by the competition organizers and thus mostly started at the same point or same phase for each shape type. We recommend keeping the wave shapes that way to preserve model accuracy. We noticed that the data wave shapes generally start with an increasing slope of 'B' and end with a decreasing slope of 'B'.

A pre-processor function has been added to shared scripts to ensure that is the case for the input 'B' field. Note that an attempt was made to make the model robust to shifts, and we think it would work if trained with shifted wave shapes, but to ensure consistency, we would prefer it if the user inputs data that adheres to the same format as the training data set. This pre-processor is informed by current knowledge that the representation of the 'B' wave shape shifts should not change the core loss.

G. Taking the geometric mean

We recommend taking a geometric mean when averaging results from the model for accuracy improvement purposes at a specific operating point from an averaging reducing variation perspective. We noticed improvements when using a geometric than an arithmetic mean, but have not done extensive studies on its usage to improve performance. The geometric mean is from the fact that the model is multiplicative in its effects from the logarithm transformation involved.

One potential application is using multiple versions of the model and averaging the results for each point. In that case, we recommend to consider taking a geometric mean. Averaging of the model results would improve the model standard deviations, thereby reducing the 95th percentile or long tails. Shared results don't use averaging of multiple versions as we would need to submit all models, thus increasing the size, but we think it would improve some benefits in an application setting dependent on how uncorrelated the errors are.

VI. MODEL DEPLOY-ABILITY AND USABILITY

Accompanying the report, the model is shared in JSON format with accompanying data transformations implemented in Python with the help of the NumPy package for matrix algebra, making it easy to transfer to other computing platforms. The model function as shared is more usable and easy to understand by anyone with an understanding of matrix algebra or any simulation software supporting matrix algebra.

To use the model in simulation software, one would need a 24-point sampler of the 'B' wave shape. The sampled wave shape concatenated with the temperature and frequency parameters would be fed to a series of matrix multiplications (weights and bias) and threshold (IF statement) or gating for ReLU implementation; changing material would amount to changing the matrices. Note: we have not provided a sample implementation.

VII. Conclusion

The ASUMag team has developed a model that has a comparable performance to the model proposed in [1] with a lower model parameter/model size. The results from this model are used across materials with varying data sizes and compositions.

The model results and model function accompany the report as per competition requirements. Overall, our model structure is capable of arbitrary wave shape prediction. The model is explained, and its application is addressed.

References

[1] Chen, Minjie; Li, Haoran; Serrano, Diego; Wang, Shukai (2023): MagNet-AI: Neural Network as Datasheet for Magnetics Modeling and Material Recommendation. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.22726115.v1

[2] D. Serrano et al., "Neural Network as Datasheet: Modeling B-H Loops of Power Magnetics with Sequence-to-Sequence LSTM Encoder-Decoder Architecture," 2022 IEEE 23rd Workshop on Control and Modeling for Power Electronics (COMPEL), Tel Aviv, Israel, 2022, pp. 1-8, doi: 10.1109/COMPEL53829.2022.9829998.

[3] Chen, Minjie (2022). How MagNet: Machine Learning Framework for Modeling Power Magnetic Material Characteristics. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.21340998.v3

[4] "Half-normal distribution," Wikipedia, https://en.m.wikipedia.org/wiki/Half-normal_distribution (accessed May 31, 2023).

[5] Otvam, "Otvam/magnet_webinar_eqn_models: 2023 Magnet Challenge Webinar: Equation-based Baseline Models," GitHub, https://github.com/otvam/magnet_webinar_eqn_models(accessed May 31, 2023).

[6] Minjiechen, "Minjiechen/magnetchallenge: Magnet challenge," GitHub, https://github.com/minjiechen/magnetchallenge (accessed May 31, 2023).