CS330 Final Report: Maximum Principal Strain of Brain Prediction Models Based on Meta Learning Approaches

Yiheng Li
yyhhli@stanford.edu

Nov/14/2020

1 Background and Related Work

1.1 mTBI

As one of the primary causes of death, traumatic brain injury poses a global health issue and affects over 1.7 million children and adults in the United States alone [1]. Though not frequently causing death directly, repetitive mild traumatic brain injury (mTBI) can potentially give rise to accumulation of brain damage that may bring about cognitive and emotional sequelae as well as cerebral blood flow alterations [2, 3]. The risk is higher as more time elapses before diagnosis [4], which calls for better monitoring of brain injury after head impacts.

1.2 Maximum Principle Strain and Finite Element Model

To quantify the risk of TBI and mTBI, brain strain, particularly maximum principal strain (MPS), is found to be a good predictor for brain injury and assorted pathology and is widely used in TBI and mTBI research [5, 6]. There were machine learning attempts to predict either whole brain MPS (which outputs the MPS of the whole brain for every region of brain, which is also what FEA model outputs) or MPS95 (which stands for the 95% maximum value of MPS) The state-of-the-art method to calculate MPS involves finite element (FE) modeling, which usually takes hours with sophisticated simulation software and trained professional engineers. There are also some machine learning models trying to fasten this process by predicting the output of finite element analysis model, however, developed machine learning prediction models focus on singular and homogeneous dataset [7, 8], which to some extend, limited their applications.

2 Goal

In real life, there are many sources of causes of brain strain that have very different dynamics and can cause different or similar amount of MPS. That can results in either different dynamics causing similar MPS95 in different datasets, or relatively similar dynamics have very different MPS95 in different datasets. And for different sports or impact source, the distribution of MPS can also by in different scales, which is shown in (Fig. 1). This project aims to address the problem of heterogeneous datasets, which inhibits machine learning MPS prediction models from easily generalizing to new data in different datasets, and hinder the model's ability to utilize extensive training data to achieve better performance.

3 Data

3.1 Datasets Description

To explore the problem of multiple dataset and prediction models, kinematics signal data from different sources to validate the results across datasets: 1422 simulated head impacts from a validated FE model of the hybrid III anthropomorphic test dummy headform [9], and was augmented to be doubled size by mirroring simulated impacts; 184 college football head impacts collected using the original version of the Stanford instrumented mouthguard [10]; 53 reconstructed head impacts from the NFL [11]; 457 mixed martial arts (MMA) [12] data; and also 48 National Highway Traffic Safety Administration (NHTSA) car crash dummy head impacts and 272 National Association for Stock Car Auto Racing (NASCAR) car racing impacts. The brain strain of the kinematics of each impact in the above-mentioned datasets were calculated with the KTH model, which is a well recognized finite element model [13], and extracted the MPS95 (95th percentile maximum principal strain rate of the brain tissue elements), MPSCC95 (95th percentile maximum principal strain rate of the corpus callosum) and CSDM as the true values for later analysis and modeling. For this project, only MPS95 is focused.

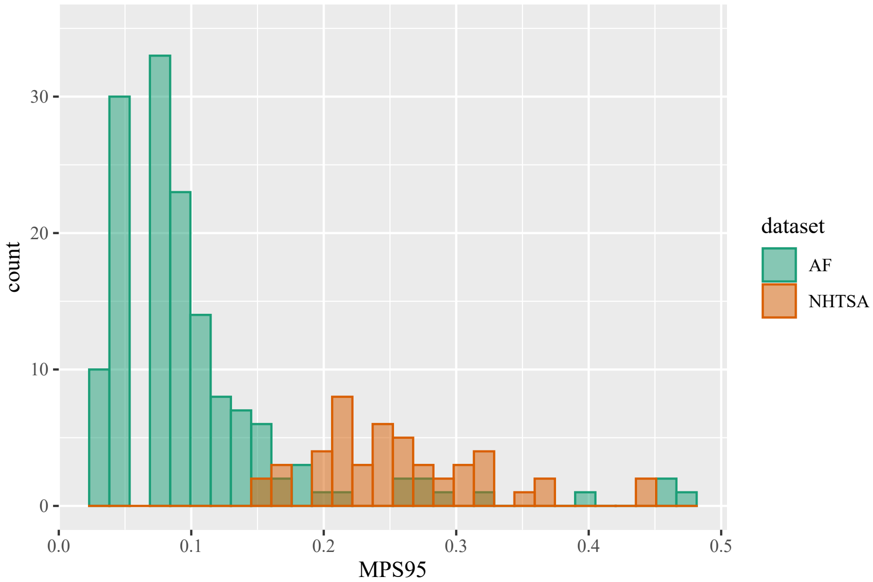

Figure 1: Histogram of MPS95 for two datasets, American Football (AF) and National Highway Traffic Safety Administration (NHTSA). The distribution of MPS95 of AF dataset is are concentrate more in values between 0 and 0.15, while the majority of MPS95 of NHTSA dataset are between 0.15 and 0.34.

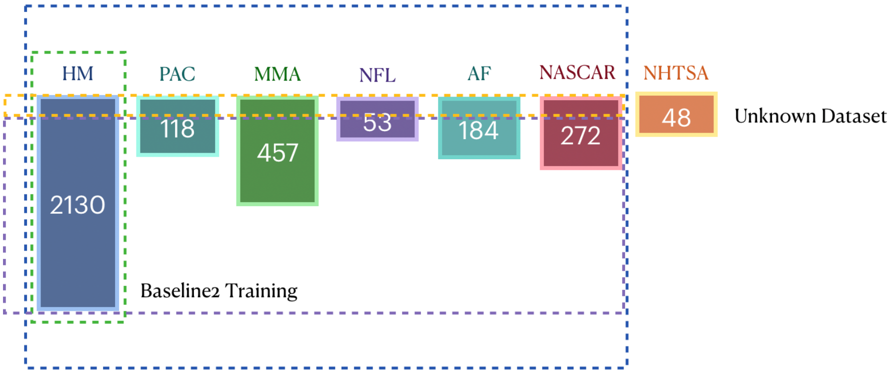

Figure 2: Data Split. The majority of six datasets are used as training and validation (purple dashed rectangle), the maximum of 10% and 20 samples of each datasets are randomly selected as test set (yellow dashed rectangle), NHTSA, which is a rather small and heterogeneous dataset is isolated as an unknown dataset to test model's performance to generalize.

3.2 Data Split

There are in total 7 datasets. For analysis purpose of this study, the datasets are divided into three parts. NHTSA is singled out to be unknown test dataset. All other datasets are divided into train and test set, which test set size to be $max\{20, 10\% \times \text{total size}\}$, as shown by Fig. 2.

3.3 Data Preprocessing and Feature Extraction

The kinematic signals of each impact consists of three spatial dimensional components of angular velocity, "X", "Y" and "Z". However, what matters more is the acceleration that the signals inferred. However, directly taking differentials is not a good practice since the signals are not "synchronized" as they are manually cropped from a sport recording. To reduce the variance of signal data, features are extracted from kinematic signal. A list of details about the features extracted was summarized in Table 1.

FFT feature details: 1-15: Mean FFT coefficient in range [0,20Hz], (20Hz, 40Hz], . . . , (280Hz, 300Hz] 16-19: Mean FFT coefficient in range (300,350Hz], (350Hz, 400Hz], . . . , (450Hz, 500Hz]

4 Method

4.1 Baseline Models

A fully-connected model that contains two hidden layers of 200 and 100 nodes followed by a dropout layer of dropout rate 0.3 are trained 500 epochs using mean square error with early stopping under the learning rate of

Table 1: Feature Descriptions

Feature Index	Feature Origin	Mathematical Processing
1-4	Angular Velocity: max(abs()) , x,y,z	power order 1
5-8	Angular Velocity: max(abs()) , x,y,z	power order 0.5
9-12	Angular Velocity: max(abs()) , x,y,z	power order 2
13-16	Angular Acceleration: max(abs()) , x,y,z	power order 1
17-20	Angular Acceleration: max(abs()) , x,y,z	power order 0.5
21-24	Angular Acceleration: max(abs()) , x,y,z	power order 2
25-28	Angular Jerk: max(abs()) , x,y,z	power order 1
29-32	Angular Jerk: max(abs()) , x,y,z	power order 0.5
33-36	Angular Jerk: max(abs()) , x,y,z	power order 2
37-40	Linear Acceleration: max(abs()), x,y,z	power order 1
41-44	Linear Acceleration: max(abs()), x,y,z	power order 0.5
45-48	Linear Acceleration: max(abs()), x,y,z	power order 2
49-67	FFT features	angular velocity in x direction
68-86	FFT features	angular velocity in y direction
87-105	FFT features	angular velocity in z direction
106-124	FFT features	angular velocity magnitude
125-200	FFT features	angular acceleration (following the previous order)
201-276	FFT features	angular jerk (following the previous order)
277-352	FFT features	linear acceleration (following the previous order)

0.0001. This model was trained using different datasets and combination of datasets. The model that trained on a training data from the mixture of all datasets are saved as the best baseline model for further comparisons.

4.2 Meta-learning

4.2.1 Memory-based Meta Learning

A memory-based meta learning approach inspired by [14] was implemented in this problem. The architecture of the model consists of two hidden LSTM layers of 128 and 32 nodes, followed by a dense layer without activation as output layer. In meta-training phase, k meta-training samples with dataset label and MPS95 embedded in the features, and in meta-testing phase, 1 sample for each dataset is also embedded with dataset label, but MPS95 is masked with 0. The model's performance is measured by mean squared error (mse), mean absolute error (mae) and root mean square error (rmse) of the meta-test set. The model is trained on training set of all datasets for 50000 epochs and tested on test sets. In default settings, the model was trained with meta-training and testing with 3 datasets with 1 training sample and 1 testing sample.

4.2.2 Optimization-based Meta Learning

In addition, a Optimization-based Meta Learning approach inspired by [15] was also implemented. To keep the consistency, the architecture of the optimization-based model is the same as the baseline model, with two hidden layers containing 200 and 100 nodes. Only the training and testing are different, as optimization-based meta learning model consists outer and inner training loops, and thus meta-learning model are trained with higher-order gradients. To be comparable, the first model is also trained with 3 datasets with 1 training sample and 1 testing sample, which for each dataset, the model learned to fit that dataset in 1 inner update, and was tested on the meta-test sample.

5 Results

5.1 Baseline Models

In the first experiment of the baseline model, Each model is trained on one dataset and tested on the same dataset or another dataset. When each model is trained and tested using training and testing data from the same dataset, the experiment is called "self-evaluation". The resulting mean absolute error is as follows: HM, 0.0129; AF, 0.0979; MMA, 0.0399; NFL, 0.2779; PAC, 0.1682; NASCAR, 0.0497. And each of the model trained were used to test other datasets' test set, which is called "cross evaluation". The result of "self" and "cross" evaluation is presented in Table 2.

From the table, the observation is that model performance varies a lot in different datasets. Some models trained and tested on single dataset showed much better results comparing with tested on different datasets. Also models performance for each dataset seem to be different because of different variance of datasets.

Table 2: Cross Evaluation Result

		Trained Data Set					
		HM	AF	MMA	NFL	PAC	NASCAR
Tested Data Set	HM	**0.0129**	0.1241	0.0914	0.1589	0.1322	0.1061
	AF	0.0943	**0.0979**	0.0969	0.2392	0.1265	0.1287
	MMA	0.0991	0.1250	**0.0399**	0.1825	0.1476	0.1054
	NFL	0.1321	0.2655	0.1398	0.2780	0.1731	0.2150
	PAC	0.1172	0.1857	0.1274	0.2945	0.1682	0.2055
	NASCAR	0.0521	0.1034	0.0649	**0.1522**	**0.1059**	**0.0497**

5.2 Meta-Learning Models

5.3 Model Comparison

Three versions of the models are independently trained and stored, and then were tested on the same batch of testing data. For two meta-learning approaches, the same meta-train meta-test data pair are used. The baseline model was trained for 5000 epochs with early stopping and stored for later experiments. The memory-based model is trained and tested for 50000 epochs and stored for later experiments with learning rate to be 2e-5. The optimization-based model is trained for 20000 epochs with inner learning rate of 1e-3, in the initial version, the learning rates are stable. The weights were initialized with keras random normal initializer. The comparison of model performance on test set for 500 test epochs is shown in Table **??**tab:comparison). From the table, we can see that memory-based meta-learning algorithm (MANN) significantly outperformed the other two methods with comparable variance, while optimization-based meta-learning algorithm is showing similar performance with baseline model. This algorithm either needs to specially tuned or is not suitable for the task.

Table 3: Model comparisons for baseline model, memory-based meta-learning and optimization-based meta-learning models. Tested on meta-test set for 200 epochs, hyper-parameters tuned for all three models, selected 5% of samples that the model performed worst on for each model and computed the worst case performance.

Metrics	Models		
	Average performance		
	Baseline	MANN	MAML
Mean absolute error	0.0420±0.0054	**0.0211±0.0093**	0.0436±0.0124
Mean squared error	0.0033±0.0011	**0.0012±0.0012**	0.0055±0.0040
Root mean square error	0.0572±0.0097	**0.0329±0.0180**	0.0720±0.0286
	Worst case performance		
Mean absolute error	0.0427±0.0057	**0.0223±0.0104**	0.0452±0.0122
Mean squared error	0.0035±0.0014	**0.0013±0.0014**	0.0061±0.0048
Root mean square error	0.0586±0.0113	**0.0350±0.0194**	0.0764±0.0311

5.3.1 Memory-based Meta Learning

Besides training the basic model, some extensive experiments are tested with memory-based meta learning model. The first one is the meta-training size is increased. The intuition for this test is that in regression cases, one sample from the dataset is not enough for the model to get enough information about the distribution of the data, so the meta-train size was tested on 1, 2, 5 and 10. From the result, we can see that for larger meta-train size, the model performed slightly worse, instead of becoming better. The performances are shown in Table 4. In addition, another experiments is changing the number of sets for meta-train and meta-test sample batches. In practice, usually what we have is one new dataset that is not seem before and the model needs to quickly converge with just a few examples as opposite to several data samples from multiple datasets. So I further trained a model for 1 dataset with 1 meta-train samples for 500 epochs, on meta-test set, its performance on test set achieved MAE of 0.0851, MSE of 0.0133, RMSE of 0.1154. Finally, I changed the model architecture from LSTM layers to GRU layers, to test the model's performance with other RNN layers, using 3 datasets, 1 meta-train sample setting , trained for 500 epochs, the performance achieved mae: 0.1518±0.0333, mse: 0.0405±0.0214 and rmse: 0.1992±0.0544.

5.3.2 Optimization-based Meta Learning

For optimization-based model, the majority of experiments are done with updating the inner learning rate as well as increase the epochs for inner update to let the model to have more flexibility to capture each set's data. First I manually updated the learning rate, with the most successful learning rates, 2e-4, 1e-3, 5e-3, the model was trained for 500 epochs, the results for this experimentation is reported in Table 5. Secondly, I increased the number of inner update epochs. Using learning rate of 2e-4 (A smaller learning rate makes more sense since

Table 4: Experimentation of different meta-training sizes for memory-based meta-learning model

Metrics	Size of meta-training set 1	2	5	10
Mean absolute error	**0.0920±0.0097**	0.1053±0.0138	0.1043±0.0104	0.1328±0.0095
Mean squared error	**0.0144±0.0039**	0.0199±0.0078	0.0180±0.0033	0.0301±0.0041
Root mean square error	**0.1196±0.0166**	0.1405±0.0283	0.1341±0.0120	0.1734±0.0118

we have more inner update epochs) to test epoch = 1, 2, 5, 10. The model performance is listed in Table 6. Lastly, I tried to implement learning the inner learning rate, but the model has no gradients over the learning rate, thus the results appeared to be the same as not updating inner learning rate.

Table 5: Experimentation of inner loop learning rate for the optimization-based meta-learning model

Metrics	Inner learning rate 2.00E-04	1.00E-03	5.00E-03
Mean absolute error	0.0485±0.0054	**0.0479±0.0052**	0.0481±0.0049
Mean squared error	0.0047±0.0010	**0.0046±0.0010**	0.0046±0.0010
Root mean square error	0.0683±0.0074	**0.0678±0.0074**	0.0676±0.0076

Table 6: Experimentation of different inner update epochs for optimization-based meta-learning model

Metrics	Inner update epochs 1	2	5	10
Mean absolute error	0.0485±0.0054	0.0484±0.0054	0.0480±0.0053	**0.0478±0.0052**
Mean squared error	0.0047±0.0010	0.0047±0.0010	0.0046±0.0010	**0.0046±0.0010**
Root mean square error	0.0683±0.0074	0.0682±0.0074	0.0679±0.0073	**0.0678±0.0074**

6 Conclusions

Memory-based meta-learning model outperforms and achieved better performance in multi-dataset scenario compared to baseline model which was trained on combined datasets.

Optimization-based model performed similarly to baseline model, which may indicates that optimization-based model is not suitable for this task.

7 Future Directions

The problem of distributional shift and multi-datasets are very common in biomedical field. Examples are data from different patients, medical image institutional difference, or medical data collected using different treatment pipelines. Meta-learning definitely have advantages in these situations, but there are still a lot more that can be explored, and the most interesting directions that I find out during the experiment from this study are, firstly, How to set up the setting for regression problems with some of the most advanced approaches, and secondly, how to deal with distributional shift and long tail distributed data specifically.

7.1 More targeted approach for distributional shift and long tail distributions

As mentioned in the final class, methods like reparametrization and adaptive network would definitely be interesting to explore. If would be good to see how application of those ideas in this dataset can make a change compared to the methods that I've used.

7.2 Meta-learning for regression problems

While regression is one of the most prevalent machine learning problems, there is not enough meta-learning approach that dig deep enough to implement their algorithms for regression. In memory-based model's publication [14], they mentioned the regression problem that they tried to solve by regressing an artificial function with varied sets hyper-parameters, which is also followed by some other researches. Though that is reasonable in the sense that the accuracy of the model is very easily estimated and the data are very easy to get, it doesn't fall into any realistic settings which we know the explicit function format that we are trying predict. And also other meta-learning methods like Prototypical-Network [16], are too specific to classification problems that cannot be used in regression questions. And another question for meta-learning to be able to used in regression scenario is that usually one sample from a dataset is not enough for the model to understand and adapt to the new dataset. Actually, many meta-learning algorithms focused on how to encode a different task (in reinforced learning) or a different class (in classification) into the model. But for regression problem, though there are ways to capture a dataset distribution, there were relatively less work targeting how to let the model to adapt to a new regression relationship.

References

[1] J David Cassidy, Linda Carroll, Paul Peloso, Jorgen Borg, Hans Von Holst, Lena Holm, Jess Kraus, Victor Coronado, et al. Incidence, risk factors and prevention of mild traumatic brain injury: results of the who collaborating centre task force on mild traumatic brain injury. *Journal of rehabilitation medicine*, 36(0):28–60, 2004.

[2] Kevin M Guskiewicz, Stephen W Marshall, Julian Bailes, Michael McCrea, Herndon P Harding, Amy Matthews, Johna Register Mihalik, and Robert C Cantu. Recurrent concussion and risk of depression in retired professional football players. *Medicine and science in sports and exercise*, 39(6):903, 2007.

[3] Colin P Doherty, Eoin O'Keefe, Eugene Wallace, Teresa Loftus, James Keaney, John Kealy, Marian M Humphries, Michael G Molloy, James F Meaney, Michael Farrell, et al. Blood–brain barrier dysfunction as a hallmark pathology in chronic traumatic encephalopathy. *Journal of Neuropathology & Experimental Neurology*, 75(7):656–662, 2016.

[4] Jonathan G Beckwith, Richard M Greenwald, Jeffrey J Chu, Joseph J Crisco, Steven Rowson, Stefan M Duma, Steven P Broglio, Thomas W McAllister, Kevin M Guskiewicz, Jason P Mihalik, et al. Timing of concussion diagnosis is related to head impact exposure prior to injury. *Medicine and science in sports and exercise*, 45(4):747, 2013.

[5] Shaoju Wu, Wei Zhao, Bethany Rowson, Steven Rowson, and Songbai Ji. A network-based response feature matrix as a brain injury metric. *Biomechanics and Modeling in Mechanobiology*, pages 1–16, 2019.

[6] Eoin O'Keeffe, Eoin Kelly, Yuzhe Liu, Chiara Giordano, Eugene Wallace, Mark Hynes, Stephen Tiernan, Aidan Meagher, Chris Greene, Stephanie Hughes, et al. Dynamic blood–brain barrier regulation in mild traumatic brain injury. *Journal of Neurotrauma*, 37(2):347–356, 2020.

[7] Shaoju Wu, Wei Zhao, Kianoosh Ghazi, and Songbai Ji. Convolutional neural network for efficient estimation of regional brain strains. *Scientific reports*, 9(1):1–11, 2019.

[8] Yiheng Li Xianghao Zhan. Fast brain strain evaluation in head impact. `http://cs230.stanford.edu/projects_winter_2020/reports/32058965.pdf`, Dec 2019. (Accessed on 10/10/2020).

[9] J Sebastian Giudice, Gwansik Park, Kevin Kong, Ann Bailey, Richard Kent, and Matthew B Panzer. Development of open-source dummy and impactor models for the assessment of american football helmet finite element models. *Annals of biomedical engineering*, 47(2):464–474, 2019.

[10] Fidel Hernandez, Lyndia C Wu, Michael C Yip, Kaveh Laksari, Andrew R Hoffman, Jaime R Lopez, Gerald A Grant, Svein Kleiven, and David B Camarillo. Six degree-of-freedom measurements of human mild traumatic brain injury. *Annals of biomedical engineering*, 43(8):1918–1934, 2015.

[11] Erin J Sanchez, Lee F Gabler, Ann B Good, James R Funk, Jeff R Crandall, and Matthew B Panzer. A reanalysis of football impact reconstructions for head kinematics and finite element modeling. *Clinical biomechanics*, 64:82–89, 2019.

[12] Douglas H Smith, Victoria E Johnson, John Q Trojanowski, and William Stewart. Chronic traumatic encephalopathy confusion and controversies. *Nature Reviews Neurology*, 15(3):179–183, 2019.

[13] Chiara Giordano and Svein Kleiven. Development of an unbiased validation protocol to assess the biofidelity of finite element head models used in prediction of traumatic brain injury. Technical report, SAE Technical Paper, 2016.

[14] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. pages 1842–1850, 2016.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[16] Sai Yang, Fan Liu, Ning Dong, and Jiaying Wu. Comparative analysis on classical meta-metric models for few-shot learning. *IEEE Access*, 8:127065–127073, 2020.