



A sensor-enabled cloud-based computing platform for computational brain biomechanics

Ritika R. Menghani^a, Anil Das^a, Reuben H. Kraft^{a,b,c,*}

^a Department of Mechanical Engineering, The Pennsylvania State University, University Park, 16802, USA

^b Department of Biomedical Engineering, The Pennsylvania State University, University Park, 16802, USA

^c Institute for Computational and Data Sciences, The Pennsylvania State University, University Park, 16802, USA

ARTICLE INFO

Article history:

Received 27 October 2022

Revised 24 February 2023

Accepted 6 March 2023

Keywords:

Brain biomechanics

Cloud computing

Nonlinear finite element modeling

Integrated mouthguard sensors

ABSTRACT

Background and Objectives: Driven by the risk of repetitive head trauma, sensors have been integrated into mouthguards to measure head impacts in contact sports and military activities. These wearable devices, referred to as “instrumented” or “smart” mouthguards are being actively developed by various research groups and organizations. These instrumented mouthguards provide an opportunity to further study and understand the brain biomechanics due to impact. In this study, we present a brain modeling service that can use information from these sensors to predict brain injury metrics in an automated fashion.

Methods: We have built a brain modeling platform using several of Amazon’s Web Services (AWS) to enable cloud computing and scalability. We use a custom-built cloud-based finite element modeling code to compute the physics-based nonlinear response of the intracranial brain tissue and provide a frontend web application and an application programming interface for groups working on head impact sensor technology to include simulated injury predictions into their research pipeline.

Results: The platform results have been validated against experimental data available in literature for brain-skull relative displacements, brain strains and intracranial pressure. The parallel processing capability of the platform has also been tested and verified. We also studied the accuracy of the custom head surfaces generated by Avatar 3D.

Conclusion: We present a validated cloud-based computational brain modeling platform that uses sensor data as input for numerical brain models and outputs a quantitative description of brain tissue strains and injury metrics. The platform is expected to generate transparent, reproducible, and traceable brain computing results.

© 2023 Published by Elsevier B.V.

1. Introduction

Brain modeling has been an active area of research in the last 40 years, with millions of dollars invested by the US government. It is difficult to determine when the first US government grant was awarded specifically for brain modeling, but notable related studies were two seminal works from Chan in 1974 and Ward and Thompson in 1975 [6,62]. Since those early efforts, brain modeling has been driven by advances in computer architectures and availability, brain imaging, experimental measurements, and injury biomechanics. In the future, the field may be further transformed by artificial intelligence and deep learning [15,65,70]. Despite the steady increase in the capabilities of computational brain biomechanics,

the modeling community has failed to transition the technology effectively into medical diagnostic tools (e.g., for physicians and athletic trainers). In addition to regulatory approvals, a primary reason for the low application of brain modeling is the lack of access and technical abilities to perform high-performance numerical simulations. Meanwhile, the modeling community has consistently prioritized linking computationally predicted metrics, such as strain, with the incidence of injury [16,20,25,64,68,70] that may be clinically transformative. The modeling community should make a better effort at expanding the usage and accessibility of their tools to fully elucidate the true value and efficacy of the tools.

Wearable sensor technology aimed at measuring head impact kinematics has advanced alongside brain modeling [24,32,50]. Recently, the use of instrumented mouthguards has been a major focus in the head injury biomechanics community. Similarly, sensor technology is also being developed to measure blast overpressure in military environments [9,45]. With the steady increase in the

* Corresponding author: Reuben H. Kraft, 320 Leonhard Building, University Park, PA.

E-mail address: reuben.kraft@psu.edu (R.H. Kraft).

number of wearable sensors for measuring head impact kinematics [4,8,12,19,23,25,27,29,33,38,40,48], the modeling community could help transform the collected sensor data into predictive measures of intracranial brain strain which in turn may be useful to medical professionals. One future potential scenario is that trustworthy wearable sensor data combined with predictive brain simulation data could be used for diagnosing traumatic brain injuries. If medical practitioners intend to combine wearable sensors and brain modeling (referred to as sensors + simulations), certain critical practical and physiological questions must be examined, including the following:

- What is the accuracy of wearable sensor data? What is the associated error?
- How can individual-specific computer models be created at scale? There could be thousands of impacts globally at a given time. As practitioners would ideally desire prompt simulation feedback, how can numerous simulations be run simultaneously?
- Can the modeling and simulation be presented in a way that practitioners find useful? Stress and strain contours are useful to an engineer, but they can be difficult to understand for non-engineers.
- What are the critical metrics to examine? Do the metrics hold any clinical relevance given the uncertainty and wide range of biovariability (material properties, geometry, injury tolerances, etc.)?
- How is the data organized for rapid identification of the most critical information?
- How can computational modeling experts increase collaboration, which will drive improvement in the quality of the models?

This paper aims to present a new open-source cloud-based platform, referred to as the Brain Simulation Research Platform, for brain modeling and simulation. The platform may help to answer a few of the questions raised above. We present a cloud-based architectural framework that can simultaneously run thousands of individual-specific brain simulations. Creating such a framework does not address questions related to clinical relevance, but the framework can be used to create large, diverse datasets (big data), where relationships might be identified and yield a capability of using sensors paired with simulations for diagnosing brain injuries.

This paper primarily focuses on the development and architectural components of the platform. In Section 2, we provide the technical details of the framework design and describe the platform in detail. We present results of using the platform in Section 3, followed by discussions and scope for future research on the platform in Section 4. Finally, we state our conclusions in Section 5. Due to the length restrictions of the journal, extensive details about the platform can be found in the Appendix. Readers are encouraged to look at Appendix C for a demonstration of the platform with a test case.

2. Platform details

2.1. Broad overview

The Brain Simulation Research Platform is hosted at <https://www.brainsimresearch.io> and was built using Amazon Web Services (AWS). The current state of the platform is a result of several iterations of computing architectures and tools that have been explored over the last few years. Fig. 1 illustrates the general system design, which includes three important pillars: a persistent interface, a persistent database for storage, and flexible or temporary computational capability.

Users can interact with the system through a persistent 'frontend' web application, from which they can interact with the data and run individual-specific brain simulations with just a few clicks. An application programming interface (API) also exists to provide a 'command-line' option for submitting finite element simulations or retrieving deep-learning and artificial intelligence predictions. The frontend web application and API server are referred to as 'persistent' (continuously operational).

To store all the data and access it quickly, the database (i.e., the 'backend storage') is also a persistent component of the system. Items such as simulation results, user data, and finite element meshes are files that need to be sent to or retrieved from a long-term storage system.

The final component is the temporary compute framework (i.e., the 'backend computing'). The cloud offers 'elasticity,' meaning computers (also referred to as instances) can be started and shut down dynamically. Elasticity saves cost and enables the platform to handle the computing load. Amazon offers many types of instances (AWS EC2) that can be used for high-performance computing.

2.2. Detailed technical description

Any study using the platform is deemed IRB (Institutional Review Board) exempt by the Pennsylvania State University as long as de-identified sensor data is used (approved via Penn State IRB STUDY00017175). The components comprising the Brain Simulation Research Platform are presented in Fig. 2, which shows that different AWS services are used for a) access and security, b) user interface and application programming interface, c) finite element model creation, d) encrypted data storage, e) simulation pre-processing, f) simulation execution, and g) simulation post-processing. The components are assembled such that the system is highly scalable and cost effective. Each subcomponent is described below.

2.2.1. Access and security

Access to the system is controlled by AWS Cognito, a service that enables user sign-up, sign-in, and access control. More details can be found in the appendix Section A.1.

2.2.2. User interface

The user interface is a progressive web application written in React.js with Redux (JavaScript Redux); hence, the interface works on desktop, tablet, and mobile hardware. Additional information about the interface is available in the appendix Section A.2.

2.2.3. Model creation

Creating a brain mesh is time-consuming because of the complexity of the structure [3,42]. An objective of this platform is to provide a process that enables automatic generation of individual-specific brain meshes without the need to manually re-create a mesh each time. This objective is achieved by first creating a 3D head surface of the user with technology developed by Avatar SDK using a picture of their face. This is followed by the generation of a custom mesh by morphing the template brain mesh to the head. The template brain meshes ("coarse" and "fine") have been created using 3D geometry from a male head model (<https://www.turbosquid.com/3d-models/male-head-3d-model/701148>, <https://blog.turbosquid.com/turbosquid-3d-model-license/>). The morphing technique uses a radial basis function [57] to fit the brain mesh to the user's head shape and size. The individual specific meshes are generated fairly quick; a user can create the mesh in a few minutes and it is then stored and can be used for any subsequent simulations for that user.

This two-step process is presented in further detail in the appendix (Section A.3). The entire process of the custom mesh generation is illustrated in Fig. 3.

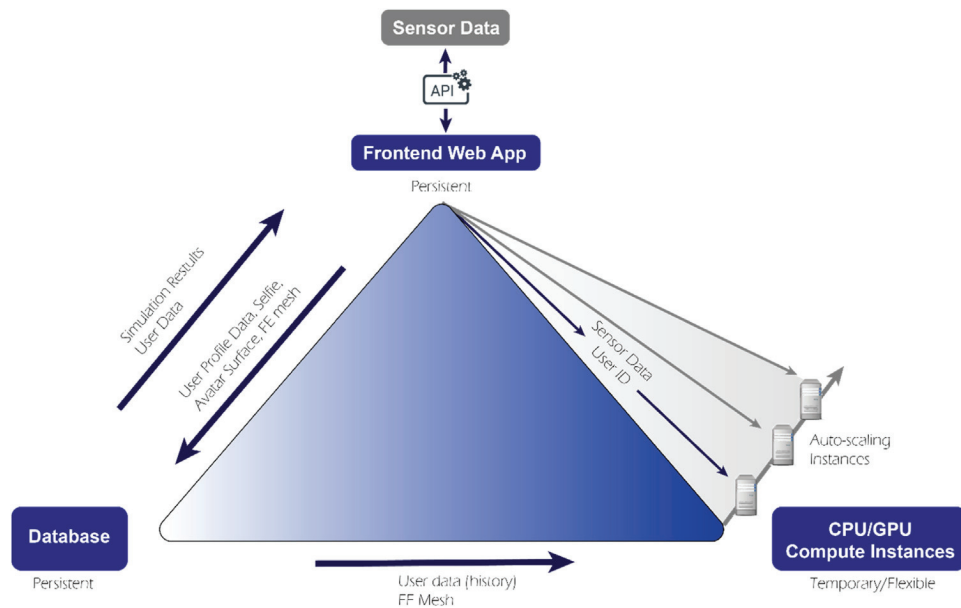


Fig. 1. Persistent and temporary framework structure. The frontend and the database are persistent, whereas the compute capability is temporary i.e. the instances are generated only when simulations are submitted. The computations are also scalable, so multiple instances can be generated at the same time if multiple jobs are submitted.

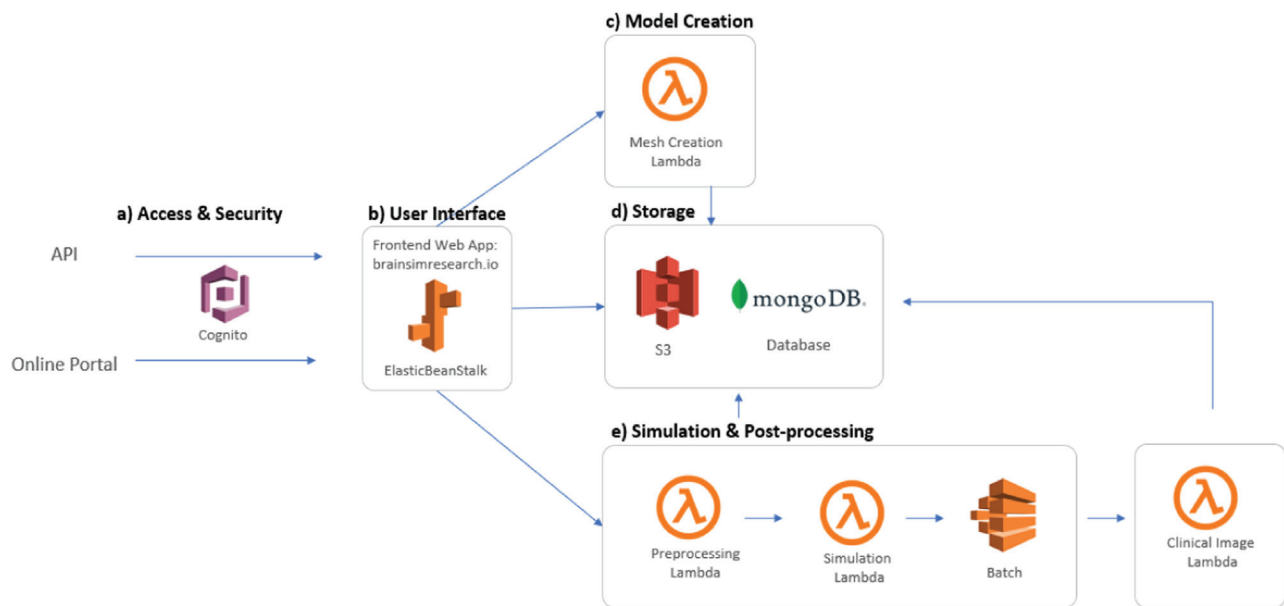


Fig. 2. System design showing different services used for different components of our platform. We use AWS cognito for user sign-in, AWS elastic beanstalk for the frontend, AWS lambda functions for different pre and post processing steps, AWS batch for job scheduling and AWS S3 and mongoDB for data storage.

2.2.4. Encrypted data storage

Owing to the potentially sensitive nature of the data, we aim to provide a platform that is compliant with the Health Insurance Portability and Accountability Act (HIPAA). Although HIPAA compliance is not claimed (nor is the system a medical device), the system employs best practices for encrypted data storage and database tools that enable rapid data access and reporting. We use MongoDB to manage the database for the platform. Additional details about data storage for our platform are presented in the appendix (Section A.4).

2.2.5. Pre-processing of simulations

When a simulation is submitted, several actions are taken prior to running the simulation. First a check is performed to identify an existing simulation: each impact is assigned a unique ID using the

sensor ID and the time of impact which is resolved down to milliseconds. These impacts have unique time traces of accelerations that are used as boundary conditions. The platform uses serverless lambda functions to determine if a directory exists for that impact based on this ID. If no directory exists, one is created for the impacts to store the meshes for the user and all the relevant results. Based on the results of the check, users can choose to either cancel that simulation or overwrite the results.

Next, the size of the job list is checked. If the size is less than 300 simulations, then serverless lambda functions are used to further preprocess and submit jobs. If the size is greater than 300, then a batch process is created to monitor job flow. If an account ID exists, then the mesh is identified and used in the simulation; if no account ID exists, then a new account is created. At the time of submission, the user can upload a profile picture of an individual

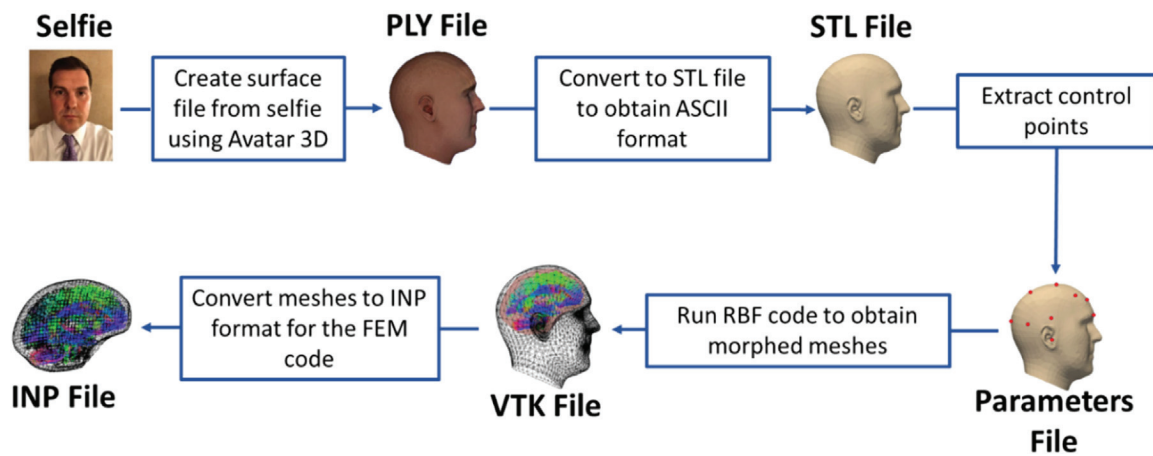


Fig. 3. Process of generation of custom brain mesh: once user uploads a selfie, it is first converted to a PLY file and then an STL file. A python script is then used to extract control points from the STL file and generate a parameters file. The parameters file is used as input for the Radial Basis Function (RBF) code and a custom brain mesh is created for the user and saved in the VTK file format. We convert the VTK to an INP format which can then be directly used in our finite element code.

wearing the sensor, and a custom finite element mesh is created and used in the simulation. Once all the checks are performed, a custom input file is written based on the sensor brand and simulation type. Finally, the mesh is saved, and the initial file is submitted. Custom materials files can also be specified, but all simulations are currently assigned the same material properties.

2.2.6. Simulation execution

The calculation of brain response and injury is carried out on a non-linear finite element code that has been developed for this platform. This code was developed using parallel computing techniques to improve runtime. Details about the formulation and implementation of the code can be found in [Appendix B](#). The finite element code can take kinematic boundary conditions as well as pressure conditions. This enables overpressure simulations [7,55,56] to be conducted as well. In order to easily build the code on any instance, docker containerization along with AWS's Elastic Container Registry (ECR) are used. This enables a cross platform computing environment that can be used on many different types of compute instances.

AWS batch processing is used to automatically allocate instances to jobs when running multiple simulations at the same time. This enables scalability of the platform as AWS does the dynamic load balancing to efficiently run several simulations at once. The platform employs both 'on-demand' and 'spot' instances to further balance computing time versus cost. More information on containerization and batch processing is available in the appendix Section A.5.

2.2.7. Post-processing of simulation

To avoid moving large files across the cloud infrastructure, we use the simulation batch instance to post-process the results. Once the results are obtained in the desired format, they are transferred for storage into S3, a storage service offered by AWS. The results are then dynamically shown on the front-end web app for the user to visualize. Post-processed results are also available via the API. Further details about post-processing can be found in the appendix (Section A.6).

3. Results

As this platform relies on a custom-built finite element code, results are presented pertaining to its accuracy and performance.

3.1. Verification and validation

We rely on a new open-source finite element code, FEMTech (<https://github.com/PSUCompBio/FemTech>), and a new set of open-source brain and skull finite element meshes (<https://github.com/PSUCompBio/brain-meshing>). Hence, our verification and validation (V&V) strategy is critical to examine.

For verification, we compared single and simple multi-element finite element problems with commercially available and widely accepted codes, such as Abaqus and LS-DYNA. We compared the models in tension, compression, and shear for relevant material models. Our verification results are shown on the Brain Simulation Research Platform website at <https://brainsimresearch.io/verification>.

Our validation strategy examines pressure, relative displacements between the brain tissue and skull, and intracranial strain using a quantitative comparison technique (i.e., Correlation and Analysis (CORA) [58]) to test the predictive capability of the model. The recommended settings for each metric [17] were used to evaluate CORA ratings. Each validation metric is presented in detail in the appendix Section B.5. Summary CORA scores for all metrics are presented in [Fig. 4](#) and [Fig. 5](#). For all validation cases for both meshes, the CORA ratings were above the marginal 0.26 value, and most of the cases were in either the 'fair' or 'good' categories.

3.2. Processing speed on AWS

The computation time of a simulation can differ significantly depending on the type of instance and number of processors used. To compare the simulation speed across different instances offered by AWS, we conducted a speed benchmarking test. [Fig. 6](#) illustrates the computation time to run a sample job on different AWS instances while using a different number of virtual CPUs (vCPUs). This sample job has a total run time of 49.6875 ms (milliseconds) and uses the coarse brain with reduced integration elements. The estimate time step for the simulation using an explicit dynamics solver is 1.78×10^{-6} s. It should be noted that in we are only considering the actual runtime for the simulations; the pre and post-processing times are excluded.

The instances tested were C6g (AWS Graviton2 based on 64bit Arm), C6i (3rd generation Intel Xeon Scalable processors), C5 (2nd generation Intel Xeon Scalable processors), C5a (2nd generation AMD EPYC processors) and C5ad (2nd generation AMD EPYC processors with additional local SSD storage). The secondary portion of the instance name contains an "nxlarge", which indicates that

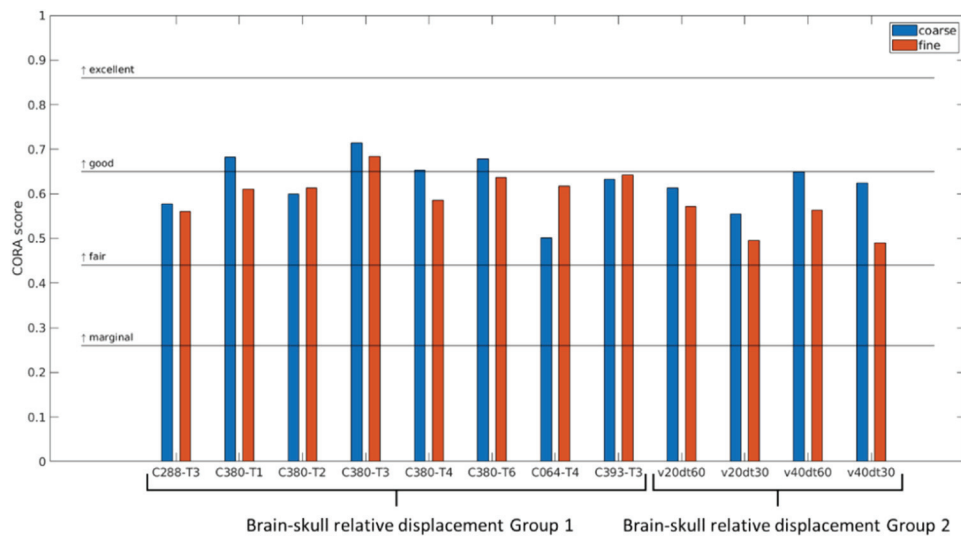


Fig. 4. CORA scores for the fine and coarse brain meshes for the brain-skull relative displacements cases from groups 1 and 2.

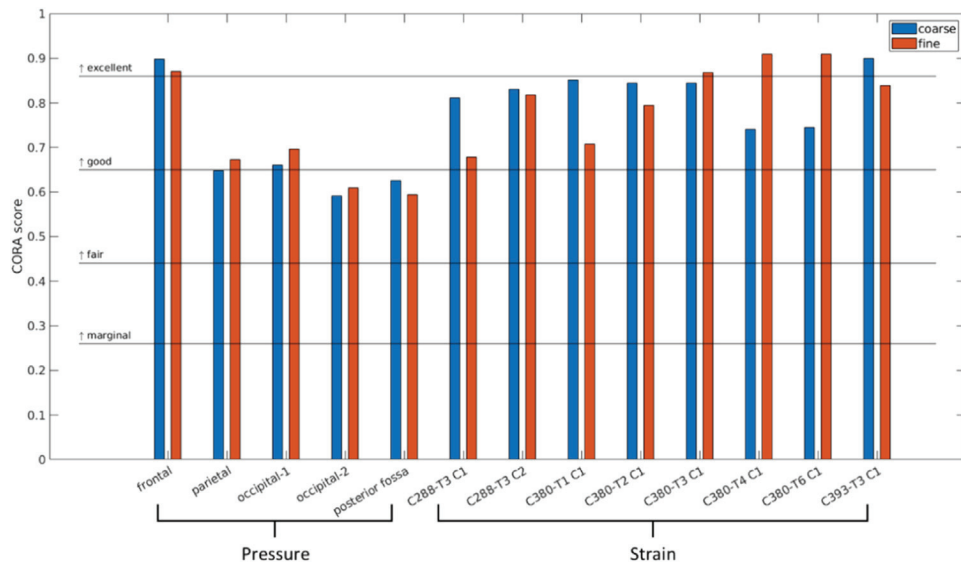


Fig. 5. CORA scores for the fine and coarse brain meshes for intracranial pressure and brain strain validation cases.

the number of vCPUs (threads) used are “n” times 4 (this number corresponds to the “large”); therefore, a 4xlarge instance uses 16 vCPUs. All these instances are recommended by AWS for compute-intensive workloads.

The different instance types can be compared by observing their performance while using 4xlarge instances. C6g.4xlarge takes the shortest amount of time: ~24 s, whereas C5ad.4xlarge takes the longest amount of time: ~39 s. An observation of the C6g and C5a instances shows that when using 4 vCPUs, the job took longer to run on the C6g instances; however, when comparing the run-times of using 16 vCPUs, the job on C6g ran faster than that on C5a. Hence, depending on how many vCPUs are being used, the optimal instance type may change.

When comparing the performance of one instance using a different number of vCPUs, as expected in most cases, the runtime reduced when the number of vCPUs was increased. Although there was significant improvement when moving from 4 C6g vCPUs to 16 C6g vCPUs (~3.5 times), moving from 16 to 64 C6g vCPUs yielded a smaller improvement (~1.2 times). Meanwhile, in the case of C5a instances, when the number of vCPUs was increased from 4 to 16 and then from 16 to 64, similar improvements were observed

(~2.4 times). However, increasing from 64 to 96 vCPUs for C5a instances yielded an increase in runtime. We suspect that this occurred because of the increase in the communication time between vCPUs. Hence, simply increasing the number of vCPUs does not always lead to faster simulations.

We also ran the same test case on Pennsylvania State University’s high-performance computing system ROAR using 16 2.8 GHz Intel Xeon processors on one node; the job took 34 s to run. Averaging all four AWS processor runtimes corresponding to 16 vCPUs, we obtained approximately 30 s, which is comparable to ROAR.

3.3. Accuracy of avatar 3D surfaces

We have conducted a study involving 24 participants to test the accuracy of the head surfaces generated using Avatar 3D. First, we needed a reference surface to compare with. This reference head surface was created by generating a point cloud of the head using laser scanning. Avatar 3D head surfaces were then created for the same users. Both surfaces were transformed to align in the same reference frame (see Fig. 7).

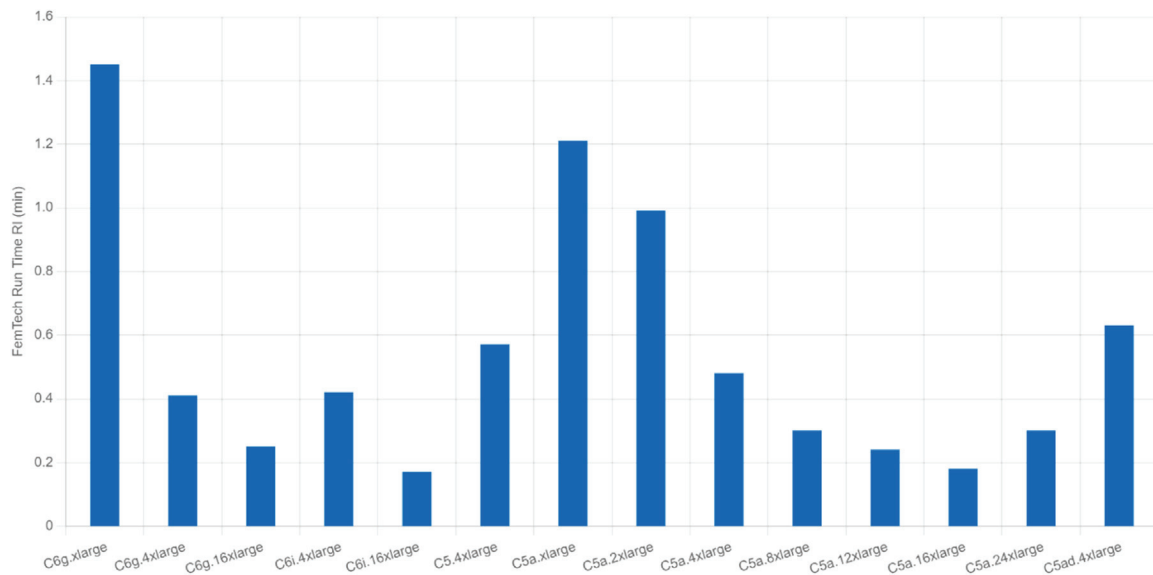


Fig. 6. Time taken to run sample job on different instances and using different number of vCPUs. Overall, the speed improves when using more processors except in the case of C5a instances when going from 64 to 96 vCPUs. On average when using 16 vCPUs, the job takes ~30 s to run.

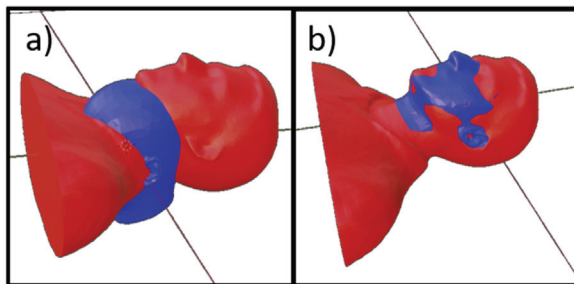


Fig. 7. a) Unaligned meshes b) Aligned meshes.

Next, a basis was required for comparison of both surfaces. For 3D mesh comparison, this means ensuring that there is some one-to-one correspondence between the two meshes. There are several ways to generate this correspondence; we opted for the icosphere mesh projection technique to create corresponding vertices on both meshes.

Once the basis was created we compared both surfaces using the Standard Hausdorff Distance (SHD). The Hausdorff distance equation for a point p on the AvatarSDK mesh given mesh M which represents the scanned mesh is shown in Eq. (1). The algorithm searches the entire mesh M for point p' which is the closest point to p based L2 norm (Euclidean distance).

$$SHD(p, M) = \min_{p' \in M} \|p - p'\| \quad (1)$$

Fig. 8 shows the results for each participant based on what percentage of vertices were within 5 mm, 10 mm and 15 mm. On average, 57% of vertices were within 5 mm SHD, 78% of vertices were within 10 mm SHD and 87% of vertices were within 15 mm SHD.

4. Discussion

Given the prevalence of head injury in contact sports and the military, TBI has become a serious concern. This has led to the development of sensors that make it possible to collect impact data and use that data for predicting brain injury. To automate the process of obtaining computational brain injury results, we created The Brain Simulation Research Platform that is a cloud-based brain

modeling service that can simulate any head impact event using impact data collected by sensors. The platform is open source to allow for collaborations with other researchers. The platform combines technology from brain biomechanics, wearable sensors, and cloud computing. Key features of the platform include:

- A custom explicit dynamic nonlinear finite element code that requires no licensing. The code is built with message passing interface (MPI) to enable parallel computing.
- A verification and validation pipeline that enables brain simulation regression testing as code changes are made.
- Individual-specific automatic custom brain mesh generation based on Avatar technology.
- Scalable computing i.e., multiple simulations can run simultaneously. We have successfully run 50,000 simulations in about 24 hrs.
- A user-friendly frontend dashboard that shows post-processed results in an organized format. Reports of the results can also be downloaded.
- The ability to read data from different sensor companies including Hybrid III data format.
- A machine learning toolkit that trains models based on simulation results and automatically computes deep-learning prediction for each simulation.

It must be noted that a similar open-source platform that also models the brain exists but deals with brain network models [52] instead, making this the first platform of its kind.

Given that our platform cuts across multiple research areas, it is tricky to draw comparisons with other research in the field. Hence, we section our discussion to enhance our overall remarks about our platform as well as to compare our work with other research.

4.1. Custom mesh generation

Research suggests that anatomically accurate brain models have better biofidelity [34]. Brain meshes for FEM simulations have been created using several different techniques: using the average of MRI (Magnetic Resonance Imaging) scans of several brains [44] or MRI scans of an individual with a healthy brain [2,10,14,53,69] by a voxel-based mesh generation technique or by using morphing techniques to convert a template brain mesh into a subject specific

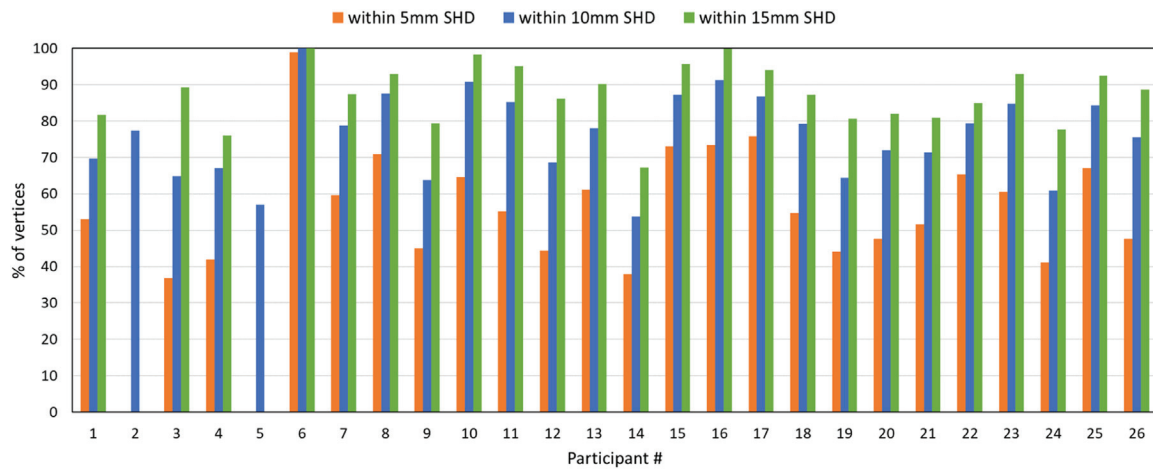


Fig. 8. Bar plot showing the percentage of vertices within 5 mm, 10 mm and 15 mm Standard Hausdorff Distance (SHD).

brain mesh [18,37,43,46]. Given the demands of our application, we used the second technique for creating user-specific meshes. We first create a head surface for each user with technology from Avatar 3D and then morph the template brain mesh to fit this surface. The term "skull adjusted brain mesh" can be used to describe this process. This terminology accurately reflects the fact that a generic brain model is being adjusted to fit the shape of a user's skull, as outlined in the avatar processing in Section 2.2.3. The assumption that the brain shape follows the skull shape is based on previous studies [63], but it warrants further research.

4.2. Computational model

Unlike several other research groups that use commercial tools to run brain models, we created a physics-based nonlinear finite element analysis code to calculate the brain's response to impact events. This was essential to avoid paying license fees incurred by using commercial tools for large-scale simulations. Moreover, creating our own code provides us with the option to customize our model with desired features.

Our model uses an Ogden hyper-viscoelastic material model for the brain, with parameters from literature [30]. Before selecting this model, different material models and parameters were tested on validation cases. Similar validation results were also obtained using a Kelvin–Maxwell viscoelastic material model with parameters from the SIMon model [54]. However, the Ogden model with viscoelasticity has a more straightforward implementation. We also considered the skull as a rigid body, enabling us to apply the input sensor accelerations to the CG for determining the skull motion.

Along with the code, we also created two types of brain meshes: a 'coarse' mesh that contains 17,030 elements and a 'fine' mesh that contains 105,520 elements. Both meshes have hexahedral elements and a mesh quality measure of scaled Jacobian greater than 0.2. The element scaled Jacobian distribution for both brains is shown in Fig. 9 and Fig. 10. The corresponding mesh volume for each interval is listed on top of the bars.

The original geometry for the skull and brain components were provided by the royalty free library from TurboSquid (<https://www.turbosquid.com/3d-models/male-head-3d-model/701148>, <https://blog.turbosquid.com/turbosquid-3d-model-license/>) which was modified to improve finite element meshability. The models include cerebral hemispheres, cerebellum, corpus callosum, basal ganglia, ventricles, brain stem, hippocampus, and thalamus. Additional layers for CSF and a skull were also added. The fine mesh captures the sulci and gyri. In addition, the Allen Human Brain Reference Atlas (<https://community.brain-map.org>)

(<https://www.brain-map.org/atlases-and-data/allen-human-reference-atlas-3d-2020-new/405/13>) was also meshed with all hexahedral elements.

To improve confidence in our code and meshes, we performed several validation tests, details of which are presented in Section 3.1, appendix Section B.5 and on our website (<https://brainsimresearch.io/validation>). We attempted to validate our model with all the recommended validation studies and even included strain rate validation on our website using the data from literature [77]. However, we did not include the details here because the CORA ratings were not calculated, as no recommended CORA settings are yet available for strain rate. We also intend on validating our code with data from live MRI tagging experiments [13,31]. To reproduce the V&V results included here, readers should download the 'Paper1' branch of FEMTech, which includes a directory FEMTech/examples/V&V containing all input files required to reproduce the V&V results shown here.

V&V is a continuous process that evolves over time. For example, as new experimental data becomes available, the current codebase should ideally be evaluated again for the new data, and previous validation data should be reexamined. Furthermore, when changes are made to the codebase, the V&V suite of tests needs to be reanalyzed. The Brain Simulation Research Platform aims to provide a transparent and reproducible V&V process that can easily be re-computed.

4.3. Machine learning

More recently, there has been an increase in efforts to apply artificial intelligence (AI) in brain response predictions [15,49,60,66,71]. These methods are advantageous because they provide rapid predictions in a fraction of the time taken for a full three-dimensional finite element model. However, all AI models are based on an underlying dataset of finite element models. Improved AI models also require robust (diverse, large, unbiased) finite element datasets. Our platform makes it possible to collect and process data for thousands of impacts and use the processed data to create datasets for machine learning.

We are working on adding machine learning capabilities to the platform; currently, we have a trained but untested model. We plan to test the accuracy of the model using the results obtained from the FEM simulations. All the results from the testing will be available on the website.

Once this feature is deployed, users will be able to submit simulations in the same way that they would for a finite element simulation; the only difference would be the need to select the machine learning option. The results will also be displayed in the

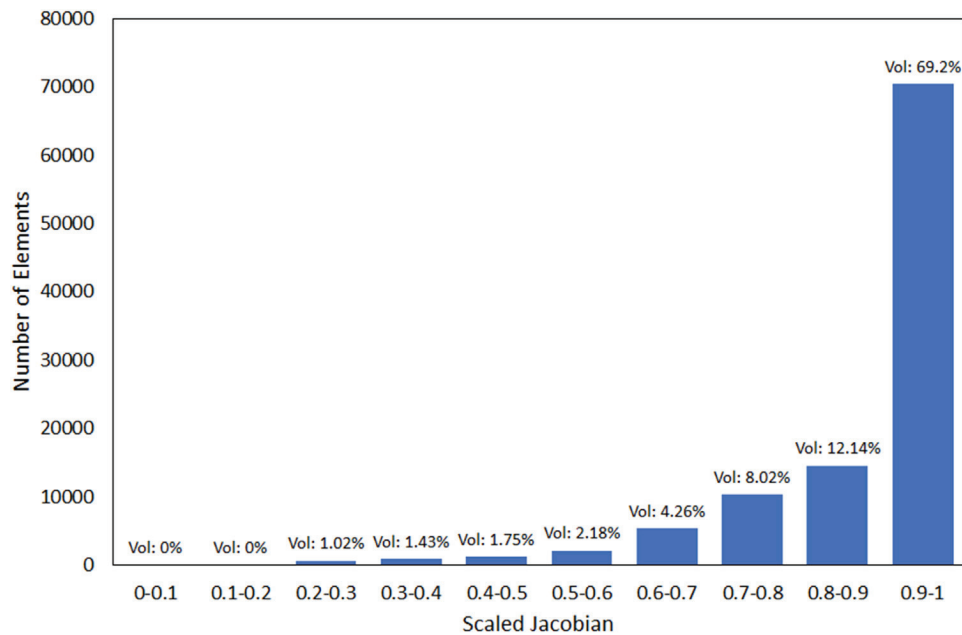


Fig. 9. The histogram shows the scaled Jacobian distribution of the elements in the fine brain mesh in intervals of 0.1; over 90% of the elements have a scaled Jacobian over 0.7. Further, the percent volume of mesh for each interval is listed on top of each bar; 69.2% of the mesh volume has a scaled Jacobian of greater than 0.9.

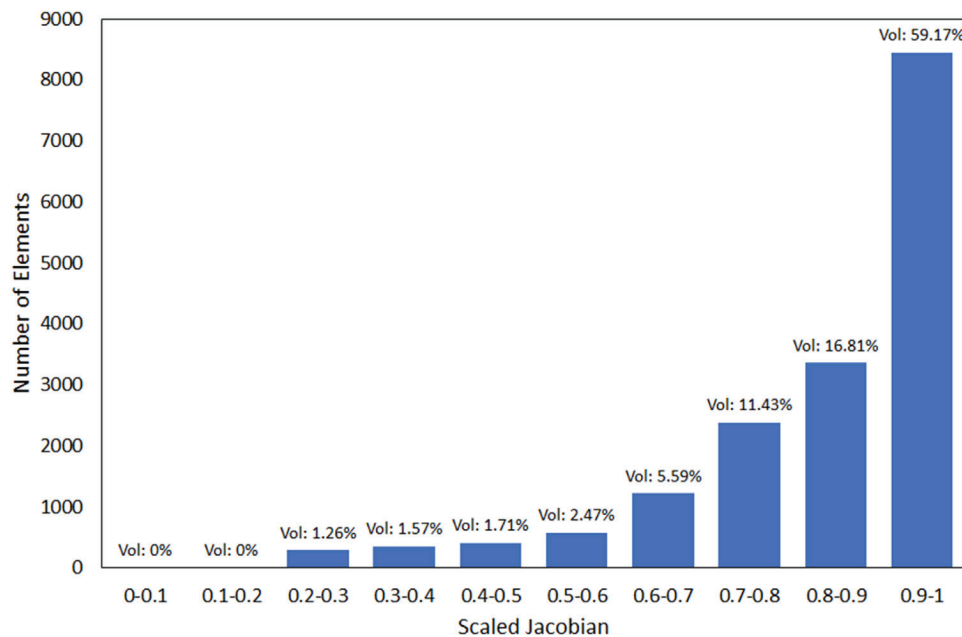


Fig. 10. The histogram shows the scaled Jacobian distribution of the elements in the coarse brain mesh in intervals of 0.1; over 90% of the elements have a scaled Jacobian over 0.6. Further, 59.17% of the mesh volume has a scaled Jacobian of greater than 0.9.

same way that they are currently displayed. From the user's perspective, the only change would be the time taken to run the simulations.

4.4. Limitations

There are limitations of the current work. Mesh convergence is an ongoing effort with the soft nonlinear materials of the brain. Convergence can be tested on a number of different variables including displacements, pressures, and strains. Similar to another study [73], 95th percentile MPS (MPS-95) has been used to test convergence; different brain resolutions have been tested for one of the validation cases and from the preliminary results it appears

that the MPS-95 does not converge for all loading conditions. This is being investigated further; more details on this study can be found in Appendix section B.7.

The accuracy of the brain meshes is yet to be determined; although the meshes are a good fit for the head generated by the avatar-based geometry creation, we only use a limited number of control points to create the custom meshes. As the custom meshes are created from the template meshes, features such as the size and shape of the sulci and gyri are based on the size and shape of the sulci and gyri in the template mesh and not on the individual's brain surface. To determine if the morphed meshes are a good representation of an individual's brain, another study will be necessary to compare the morphed meshes and MRI scans of the

same individuals. In future work, we aim to increase the personalization of computational brain models.

Another limitation is that our current brain model uses only an isotropic material model. We only have the implementation of the Ogden and Neo-Hookean models along with viscoelasticity; we currently do not have the capability to include anisotropy. In the near future we plan on including embedded elements in order to model fiber tractography which will add anisotropy to the models. Furthermore, our code works with full and reduced integration hex elements; implementation for shell elements to include falx and tentorium is still underway.

5. Conclusion

We present the Brain Simulation Research Platform, an easy-to-use service that can compute brain response and injury. The use of high-performance cloud computing makes the platform accessible to any number of users with no capacity limit. By adding the functionality to directly input sensor data, we can predict brain injury caused by real-world impacts in a reasonable amount of time. We currently use finite element modeling as our basis for predicting injury; however, we are working on adding machine learning capabilities to reduce the computation time significantly. As the platform is already being used to run large-scale simulations, we would like to invite other research groups to use and contribute to the platform as well. With this platform, we hope to be one step closer to using brain models in clinical applications.

Declaration of Competing Interest

Reuben Kraft has a financial interest in BrainSim Technologies Inc., a company which could potentially benefit from the results of this research. This interest has been reviewed by Penn State University in accordance with its Individual Conflict of Interest policy for the purpose of maintaining the objectivity and integrity in research and is being managed by Penn State University.

Acknowledgements

Most of this work was supported through the National Science Foundation CAREER award under award number [1846059](#). Any opinions, findings and conclusions expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation. Early parts of the work were also funded by a subcontract from SURVICE Engineering Company, LLC which was under a prime Department of Defense Prime Contract Number DOTC-17-01-INIT0086, OTA 2014-322. R.H.K would like to thank Mr. Christopher Griffin of Amazon Web Services for assistance obtaining research credits on Amazon Web Services. In addition, throughout this process, we have worked closely with various sensor companies to understand their needs, understand the way they store data, and how they envision using brain simulations in the future. We are deeply grateful for their collaboration and help.

Appendix A. Additional details of platform architecture:

A.1. Access and Security

AWS Cognito scales to millions of users and supports sign-ins with social identity providers, such as Apple, Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0 and OpenID Connect. Once a user is created, they are assigned an account ID that is entered into Cognito as well as a 'relational' database.

User permissions are also a critical part of the platform, and the permission construct is organized according to organizations,

institutions, teams, groups, and individuals. The permissions were originally designed for principal investigators (PI) of research organizations. A PI who runs a research group would often wish to grant different levels of data visibility to members of their research group. In addition, the PI may be running multiple studies at different institutions, and the studies would require compartmentalization and security. Thus, the PI would be assigned a 'Level 300' permission for their organization. The PI would be able to assign another Level-300 admin who could manage the data organization for the research laboratory.

A.2. User Interface

The web application runs on AWS Elastic BeanStalk, which enables scalability as additional users become active simultaneously. This means additional instances running the web application are dynamically created when the CPU usage becomes greater than 80%. In addition, we have an API that would be useful for groups working on sensor technology, allowing for a seamless transfer of sensor data to the platform.

A.3. Model creation

First, a three-dimensional surface model of an individual is generated using a two-dimensional picture of the individual's face. This model is generated with the help of Avatar3D, a software toolkit that uses machine learning algorithms and machine vision. [Fig. A.1](#) shows an example of surfaces generated using the technology. The platform uses the version without hair for processing.

Next, the radial basis function approach [\[35,36,61,72\]](#) is used to map fine and coarse template finite element brain meshes (see [Fig. A.2](#)) to the target head geometry (in this case, the target is the head surface without the hair). The radial basis function interpolates the coordinates of the brain mesh for the target surface and provides a morphed mesh. Control points are selected in regions that define the structure of the head. The control points currently used can be adjusted if required; however, preliminary results are promising.

A.4. Encrypted Data Storage

As the simulation data is analyzed to present the user information back on the frontend, a database is important to search and query the data. Through experimentation with different types of databases, we found that a 'relational' database is best suited for the platform. We initially used a non-relational database but found that conducting queries to show data on the frontend severely slowed the responsiveness of the platform. This is because we would repeatedly look up one value from a table to get data from



Fig. A.1. Head surface files created by Avatar SDK; one includes hair and shoulders, the other is a hairless version and only shows the head. The hairless version is more accurate and is therefore used for the generation of custom meshes for users.

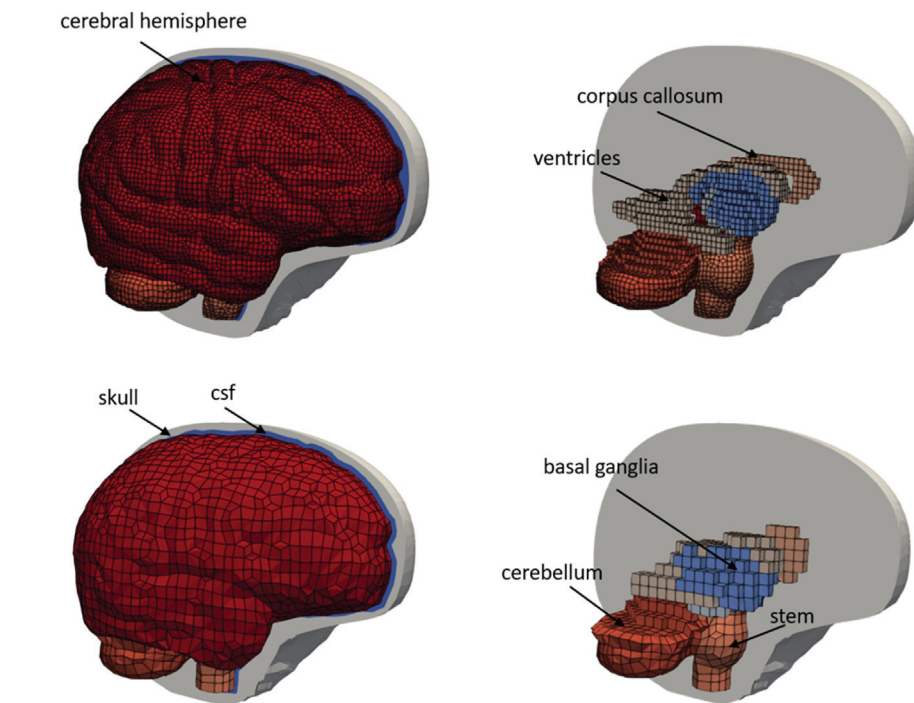


Fig. A.2. The fine brain mesh (top) and the coarse brain mesh (bottom). A cut view shows different parts included in the meshes. These can be assigned different material properties if needed.

another table. By switching to a relational database, we created pre-existing connections between collections of data, thereby making the queries faster. The different connections are formed by how data is queried on the frontend and what type of data can be sent into the system.

The relational database schema is presented in Fig. A.3. The schema defines the relationship between collections of data; each outline in the figure is referred to as a collection. A closer observation of the 'sensor details' in the angular-acceleration collection shows that the collection contains a considerable amount of the information sent to the system when a simulation is submitted. Connections can be observed between sensor details and other data collections with arrows indicating either 'virtual relation' or 'object'. The virtual relation arrows indicate connections that exist across different collections of data; for example, a virtual relationship is observed between 'sensor details' in the 'angular-accelerations' collection and 'users' in the 'user-sensors' collection. The object arrows link to data belonging to the same collection but is presented separately for clarity; for example, 'players' is linked with 'sensor details' and both belong to the angular accelerations collection. A careful observation of all the virtual relations reveals how different collections are connected. Different configurations of relations and collections could be made; however, the current format has been tested extensively.

The relational database used for the platform is MongoDB, which has different pricings for different numbers of connections per second to the servers. This is an important feature as simulations are scaled. For example, we have observed that for a single simulation, as many as four connections to the database servers may exist at a given instance of time, depending on several factors. The goal is to minimize the number of simultaneous database connections to reduce cost. The number of database connections could be 4000 for 1000 simultaneous simulations. Therefore, care should be taken when 'bursting' simulations—if a sufficient number of database connections are not paid for, errors could be generated in the system.

A.5. Simulation Execution

Containerization:

Owing to the complex nature of our code, building can be delicate and time-consuming. Several dependencies of the code require installation, and not all the dependencies are available on all operating systems. Hence, to achieve a faster build that works across all platforms, we use Docker containers. Docker is an open-source platform that allows users to build, deploy, and run applications by packaging the code and its dependencies using images. Building our code on a docker image allows it to be deployed on any machine or even a cloud computing system that has docker installed. Running an image creates a container, which is used to run the code. We have several images for different aspects of our application; hence, to create the final docker image, we use a multistage build. This ensures that the final image contains only the files necessary to run the code.

Batch computing:

Amazon offers a batch computing service (AWS Batch), which automatically allocates resources to jobs according to their requirement. This feature enables users to submit several simulations at once on the platform, and the jobs receive instances based on predefined parameters. When multiple simulations (over 300) are uploaded, our platform first creates a serverless compute function that preprocess all the files to identify accounts and group events together; at this point, the user can choose to remove any simulation that may have already been run. Submitting the simulations then spawns a batch job that divides the simulations across different job queues that are already set up. Currently, five queues are set up, and each queue can handle 300 jobs. Once the first queue reaches 300 jobs, the 301st job goes to the second queue, and so on. This process helps to reduce wait times for the jobs. When the user uploads less than 300 jobs, the jobs are automatically sent to the job queue; in this case, no batch job is created to divide the jobs among queues.

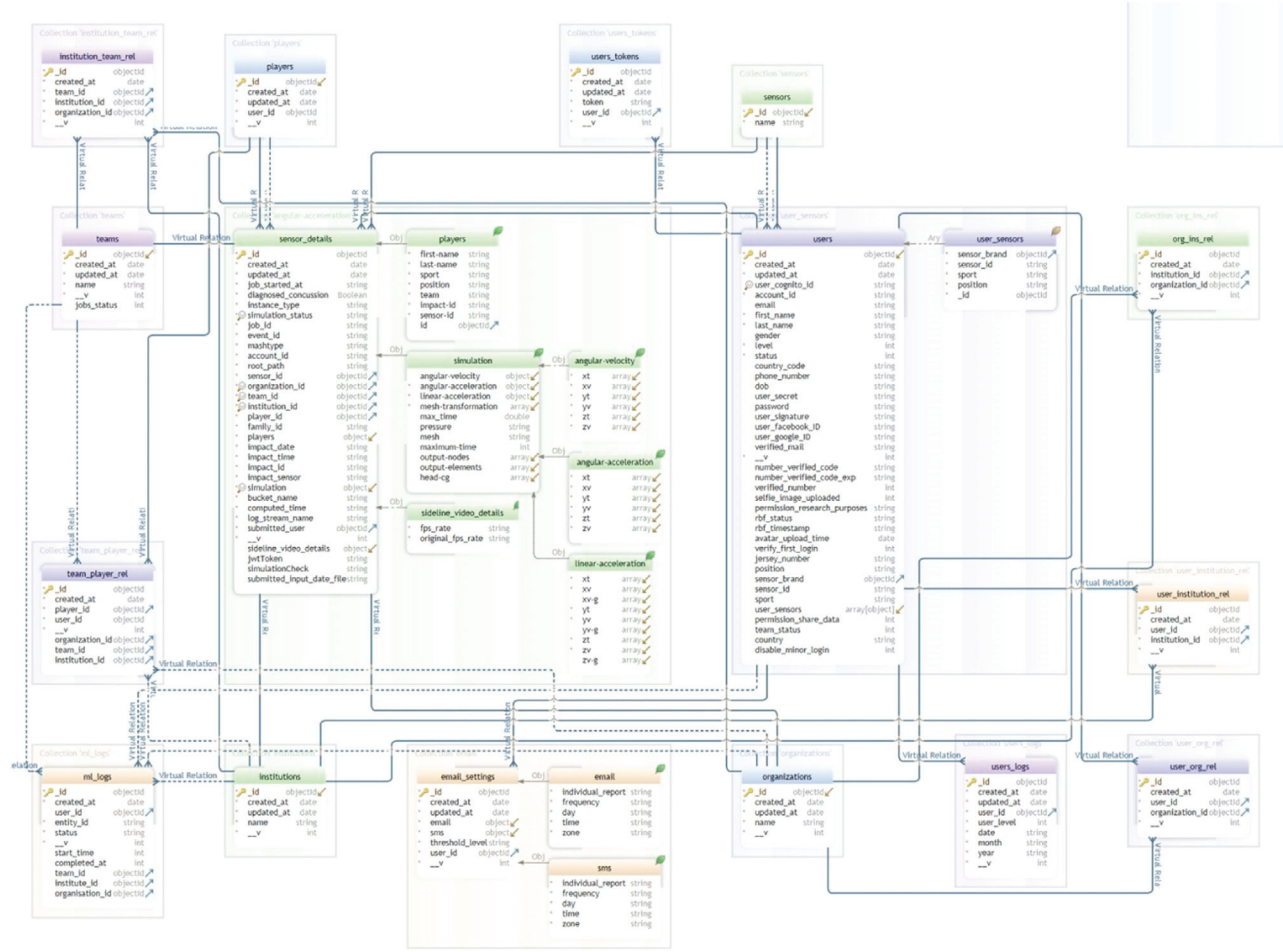


Fig. A.3. Entity relationship diagram (ERD) showing the relations between collections in our database.

A.6. Postprocessing

The post-processing stage uses several lambda functions (AWS Lambda); this is a scalable serverless compute service offered by AWS that can run a script when triggered. For example, when a new simulation is run, an output file containing all the injury metrics is generated and uploaded to S3. Once this file is uploaded, a lambda function is triggered to update a summary file for the user with the injury results from that impact. The summary file is used for displaying all results on the player dashboard.

In the near future, we plan to use step functions for post-processing. A step function is another service offered by AWS and can create a workflow for the lambda functions. This makes it possible to visualize the process and track failed lambda functions. Currently, if a lambda function fails for any reason (such as a connection error), it is difficult to track the failed function due to the number of lambda functions. Thus, using step functions enables better error handling (e.g., by adding retry logics).

Appendix B. Finite Element Code (FEMTech):

We developed a custom, open-source nonlinear finite element code known as FEMTech (Finite Element Modeling Technology) to run brain simulations. The custom code is built on message passing interface (MPI) to facilitate simulation runtime through parallel FEM calculations. Benchmarking tests were run to determine the improvement in runtime due to parallel processing; the results of the test are presented in Section 3.2.

The finite element code uses the updated Lagrangian method for an explicit dynamics scheme. Updated Lagrangian codes use equations that reference the mesh in the current time step. The finite element formulation is derived from the generalized momentum equations, including the momentum equation (equation (B.1)), traction boundary conditions (equation (B.2)), and interior continuity condition (equation (B.3)). Together, these are called the strong form equations:

$$\frac{\partial \sigma_{ji}}{\partial x_j} + \rho b_i = \rho \dot{v}_i \text{ on } \Omega \quad (B.1)$$

$$n_j \sigma_{ji} = \bar{t}_i \text{ on } \Gamma_t \quad (B.2)$$

$$n_j \sigma_{ji} = 0 \text{ on } \Gamma_{int} \quad (B.3)$$

where σ is the stress, ρ is the density, b is the body force, v is the velocity, Ω is the current domain, n is the unit normal, $\bar{t}$ is the traction and Γ is the boundary. As the strong form equations cannot be directly discretized, the weak form equation is developed.

$$\int_{\Omega} \frac{\partial (\delta v_i)}{\partial x_j} \sigma_{ji} d\Omega - \int_{\Omega} \delta v_i \rho b_i d\Omega - \sum_{i=1}^{n_{SD}} \int_{\Gamma_t} \delta v_i \bar{t}_i d\Gamma + \int_{\Omega} \delta v_i \rho \dot{v}_i d\Omega = 0 \quad (B.4)$$

This equation can also be written as the virtual power equation:

$$\delta P = \delta P^{int} - \delta P^{ext} + \delta P^{kin} = 0 \quad \forall v_i \in u_0 \quad (B.5)$$

where the virtual internal power is given by $\delta P^{int} = \int_{\Omega} \frac{\partial(\delta v_i)}{\partial x_j} \sigma_{ij} d\Omega$, the virtual external power is given by $\delta P^{ext} = \int_{\Omega} \delta v_i \rho b_i d\Omega + \sum_{i=1}^{n_{SD}} \int_{\Gamma_i} \delta v_i \bar{t}_i d\Gamma$, and the virtual kinetic power is given by $\delta P^{kin} = \int_{\Omega} \delta v_i \rho \dot{v}_i d\Omega$.

These equations can be found in nonlinear finite element modeling books [5]; however, we have added them here for the reader's convenience. The implementation of these equations is presented in Section B.4. Features of our code relevant to model validation are described in the upcoming sections.

B.1. Reduced Integration

The brain meshes of our platform are created using 3D hex elements. The most straightforward implementation for these elements is the fully integrated scheme. While it has good accuracy, shear locking (when subjected to bending loads) and volumetric locking (when using nearly incompressible materials) make it an undesirable element to use. These problems can be avoided by using an element with fewer integration points. An example is the reduced integration element that uses only one integration point. Consequently, the model runtime is significantly reduced because of the reduction in the number of calculations. This is a substantial advantage as brain simulations can often run for hours because of the small elements and near incompressible material properties involved.

The shape functions used for implementing the reduced integration element are given by the equation

$$N_j = \frac{1}{8} (1 + \xi \xi_j) (1 + \eta \eta_j) (1 + \zeta \zeta_j) \quad (\text{B.6})$$

where (ξ_j, η_j, ζ_j) are substituted as combinations of $(\pm 1, \pm 1, \pm 1)$, and (ξ, η, ζ) represents the coordinates in the natural coordinate system; because a single integration point exists, we can substitute (ξ, η, ζ) as $(0, 0, 0)$.

A disadvantage of using a reduced integration scheme is hourglassing, which is a numerical instability that causes spurious deformation modes that can quickly destroy the solution. Four modes can arise because of hourglassing in each direction (12 in total). As these displacements are artificial, they are known as zero-energy modes.

To prevent the mesh from encountering hourglassing, an anti-hourglass scheme is implemented. Hourglass shape vectors γ are used to determine the occurrence of hourglassing. If hourglassing is detected, an anti-hourglassing force is computed and added to the internal forces at the nodal level, thereby making the element stiffer. We use the implementation from the LS-DYNA theory manual [39]; more details of this scheme can be found in the original research paper [11]. To ensure that none of the rigid body, normal strain, and shear strain modes are affected by this additional force, the hourglass shape vectors are formulated to be orthogonal to the linear velocity field:

$$\gamma_{\alpha k} = \Gamma_{\alpha k} - N_{k,i} \sum_{n=1}^8 x_i^n \Gamma_{\alpha n} \quad (\text{B.7})$$

where Γ represents the hourglass base vectors (Table B.1), and x represents the coordinates of the element in the current timestep. The anti-hourglassing force is calculated as:

$$f_{i\alpha}^k = a_h g_{i\alpha} \gamma_{\alpha k}, \quad (\text{B.8})$$

where $g_{i\alpha} = \sum_{k=1}^8 \dot{x}_i^k \gamma_{\alpha k}$, $a_h = Q_H G \rho v_e^2 c / 4$, $\dot{x}$ is the nodal velocity, Q_H is a constant (typically between 0.05 and 0.15), ρ is the element density, v_e is the element volume, and c is the speed of

Table B.1
Hourglass base vectors.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$
Γ_{j_1}	1	1	1	1
Γ_{j_2}	1	-1	-1	-1
Γ_{j_3}	-1	-1	1	1
Γ_{j_4}	-1	1	-1	-1
Γ_{j_5}	-1	1	1	-1
Γ_{j_6}	-1	-1	-1	1
Γ_{j_7}	1	-1	1	-1
Γ_{j_8}	1	1	-1	1

sound in the material. If no hourglassing occurs, $g = 0$, eliminating anti-hourglassing forces.

B.2. Ogden Hyper-elastic Model

Since the brain bulk modulus is significantly higher than its shear modulus, the brain behaves like a nearly incompressible material. Hyper-elastic models are able to capture large deformation mechanics for nearly incompressible materials, making them a suitable option to model the brain. Several head models have used the Ogden material model for the capturing the brain's motion [14,26,30]. To account for the rate dependence of the brain's motion, a linear viscoelastic model is used as an addition to the Ogden model; this has been described further in the next section.

As the brain has a small amount of compressibility, the volumetric component of the strain energy equation is treated separately.

$$W = \sum_{i=1}^3 \sum_{j=1}^2 \frac{\mu_j}{\alpha_j} (\bar{\lambda}_i^{\alpha_j} - 1) + \frac{K}{2} (J - 1)^2 \quad (\text{B.9})$$

W is the strain energy function; μ , α , and n are Ogden material constants; and K is the bulk modulus; the values of these constants are listed on the website at <https://brainsimresearch.io/code-details/v1.0>. $\bar{\lambda}$ is the volumetric independent principal stretch calculated from the principal stretch λ as:

$$\bar{\lambda} = \frac{\lambda}{J^{1/3}}, \quad (\text{B.10})$$

where J is the element Jacobian. Cauchy stresses are calculated from the strain energy function using the following relation:

$$\sigma_{ij} = \frac{\lambda_1}{\lambda_1 \cdot \lambda_2 \cdot \lambda_3} \frac{\partial W}{\partial \lambda_1} b_i^{(1)} b_j^{(1)} + \frac{\lambda_2}{\lambda_1 \cdot \lambda_2 \cdot \lambda_3} \frac{\partial W}{\partial \lambda_2} b_i^{(2)} b_j^{(2)} + \frac{\lambda_3}{\lambda_1 \cdot \lambda_2 \cdot \lambda_3} \frac{\partial W}{\partial \lambda_3} b_i^{(3)} b_j^{(3)} \quad (\text{B.11})$$

λ is calculated as the square root of the eigenvalues of matrix B , and b represents the eigenvectors of that matrix. By substituting W from equation (B.9) in equation (B.11), we get the following equation:

$$\sigma_{pq} = \sum_{j=1}^3 \left[\frac{1}{J} \sum_{i=1}^2 \mu_i \left(\bar{\lambda}_j^{\alpha_i} - \frac{1}{3} (\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i}) \right) + K * (J - 1) * I \right] b_p^{(j)} \otimes b_q^{(j)} \quad (\text{B.12})$$

We then add the contribution of viscoelasticity to these stresses, the details of which are explained in the next section.

B.3. Viscoelasticity

The brain exhibits viscous behavior, i.e., the motion of the brain lags behind the motion of the skull. Thus, it is essential to include a viscoelastic model when simulating the brain. Like several other research groups, we add viscoelasticity onto a hyper-

1. Read input file and mesh
2. Transform and partition mesh
3. Initialize all variables to zero
4. Compute shape functions and derivatives of shape functions $N, dN/dx$
5. Set up mass matrix M
6. Compute initial time step Δt
7. Begin time loop and repeat till simulation end time is reached
8. Compute the different time variables

$$t^n = Time, t^{n+1} = Time + \Delta t$$
9. Apply acceleration/velocity boundary conditions (see box 2)
10. Update Time

$$Time = t^{n+1}$$
11. Calculate half velocities if nodes do not have any boundary conditions:

$$v^{n+1/2} = v^n + \frac{\Delta t}{2} a^n$$
12. Compute displacements

$$u^{n+1} = u^n + \Delta t * v^{n+1/2}$$
13. Call GetForce routine (see box 3)
14. Calculate accelerations

$$a^{n+1} = \frac{(f_{net} - C_{damp} v^{n+1/2})}{M}$$
15. Calculate velocities

$$v^{n+1} = v^{n+1/2} + \frac{\Delta t}{2} a^{n+1}$$
16. Check Energy
17. Write required output variables
18. Compute time step increment

$$\Delta t = Reduction\ Factor * \frac{Characteristic\ Length}{Wave\ Speed}$$
19. Check if simulation end time has been reached, if not, repeat from step 7

Box B.1. Explicit dynamic scheme used in FEMTech.

elastic model [10,30,59,67,74]; however, certain groups only model the brain with a viscoelastic model [41,51,54,69].

Several implementations exist for viscoelasticity; we implemented a scheme that adds three-dimensional viscoelasticity at finite strains [28]. We first calculate an internal stress variable (H) that uses information from the previous stress state and then adds it to the deviatoric stresses at the current time step. The equations for the computation of the internal stress variable are in the time = 0 reference frame; thus, we first need to convert our Cauchy stresses (σ) to PK2 stresses (S). Once we add the contribution of the internal stress variable to the deviatoric PK2 stress, we then convert it back to the Cauchy stress.

$$H_j^{n+1} = e^{-(\Delta t/\tau_j)} H_j^n + \gamma_j \left(\frac{1 - e^{-(\Delta t/\tau_j)}}{\Delta t/\tau_j} \right) [DEVS_0^{n+1} - DEVS_0^n] \quad (B.13)$$

$$DEVS^{n+1} = DEVS_0^{n+1} + \sum_{j=1}^N H_j^{n+1} \quad (B.14)$$

$$dev\zeta^{n+1} = F^{n+1} DEVS^{n+1} (F^{n+1})^T \quad (B.15)$$

$$\zeta^{n+1} = J^{n+1} \left(\frac{\partial W}{\partial J} \right)^{n+1} I + dev\zeta^{n+1} \quad (B.16)$$

$$\sigma^{n+1} = \frac{\zeta^{n+1}}{J^{n+1}} \quad (B.17)$$

The Cauchy and internal stress variables from this time step are then stored and used for the calculations in the next time step.

B.4. Finite element code implementation

Box B.1 shows the general implementation of the explicit dynamics scheme in FEMTech. Implementation of the rigid body motion and internal force computation can be found in Boxes B.2 and B.3, respectively.

B.5. Validation

For each case, the head model was scaled according to the dimensions provided in the respective reference papers. Further discussion is provided below for each metric.

1. Compute angular velocities at generator r half time step using runge-kutta stepping algorithm

$$\dot{\omega} = \alpha, \quad \dot{r} = \frac{1}{2} \omega \times r + \omega \frac{r_{mag} \cot(r_{mag})}{2} + r \frac{r \cdot \omega}{2r_{mag}^2} (1 - r_{mag} \cot(r_{mag}))$$
2. Calculate the rotor R

$$R = e^r$$
3. Calculate the new coordinates of the rigid nodes after rotation

$$V_p = R V R^{-1}$$
4. Find half velocities for nodes with rigid boundary conditions

$$v_{half} = \omega \times V_p + \text{linear velocity}$$

Box B.2. Calculation of velocity for boundary conditions.

1. Initialize internal forces

$$f_i = 0$$
2. Calculate external forces based on boundary conditions and store it under the net force

$$f_{net} = f_e$$
3. Loop over elements
 - Loop over quadrature points
 - Calculate deformation gradient and other deformation related tensors

$$F = I + \frac{\partial u}{\partial x}, B = F F^T$$
 - Calculate stresses (see ogden model and viscoelasticity)
 - Calculate internal forces

$$f_i = f_{it} + \frac{\partial N_i}{\partial x} \sigma J \xi \overline{W_Q}$$
 - Check for hourglassing and correct if needed (see reduced integration)

$$f_i = f_i + f_{HG}$$
 - Update total forces

$$f_{net} = f_{net} - f_i$$

Box B.3. GetForce routine.

B.5.1. Pressure

We first compared pressure traces from Nahum [47]. The acceleration curve for experiment 37 was used as input and was applied to the head CG. The output was calculated by averaging the pressure across a range of elements for each of five regions: frontal, parietal, occipital 1 and 2, and posterior fossa. The locations of the elements selected for the pressure calculation were based on figures showing similar elements in the validation of another brain model [54]. The overall CORA rating for the Nahum case was 0.685 (good). The comparison plots and CORA scores for each region can be found here: <https://brainsimresearch.io/pressure-response>.

B.5.2. Brain Displacements

We compared relative brain-skull motions obtained from the experiments of Hardy [21,22], referred to as group 1, and Alshareef [1], referred to as group 2. For both groups, acceleration boundary conditions were applied to the head CG. Relative displacements were computed by subtracting the rigid displacement from the output nodes, which were the nodes closest to the location of the neutral density targets (NDT)/sonomicrometry crystals in the experimental studies. Eight cases were tested under group 1: two occipital impacts, one parietal impact, and five temporal impacts; four cases were tested under group 2: all temporal impacts.

Detailed validation results for brain displacements are shown at <https://brainsimresearch.io/brain-displacement-response>.

B.5.3. Brain Strain

Next, we compared Green-Lagrangian strains to the experimental data [76]. To compute strains, a tetra structure was created according to the locations of the NDTs in the brain. All elements identified as part of this structure were then used for the strain calculations by averaging their strains [76]. An example is shown in Fig. B.1. All the brain strain validation results are available at <https://brainsimresearch.io/brain-strain-response>.

B.6. Benchmarking

We have compared our mesh validation results (Table B.2) with other brain models that use the same CORA parameters; these include the ADAPT model [34], original KTH model [30,34] and the WHIM anisotropic model version 1.5 [75].

In addition, we are including two sample test cases that we tested on our models; these can be used for benchmarking other brain models. Both cases use a morphed brain mesh that is based on dimensions of the 50th percentile male head. The boundary conditions were applied to the CG with the skull kinematically coupled to it. For both cases the results presented are the 95th percentile MPS (MPS-95) curves in seven brain regions: Motor and

Table B.2
Comparison of FEMTech CORA scores with other brain models.

Test	Case	CORA scores				
		FEMTech Coarse	FEMTech Fine	ADAPT	KTH	WHIM (v1.5)
Displacement High Rate	C064-T4	0.504	0.618	–	–	–
	C288-T3	0.577	0.561	0.588	0.423	0.696
	C380-T1	0.683	0.611	0.694	0.56	0.693
	C380-T2	0.599	0.614	0.549	0.416	0.588
	C380-T3	0.715	0.684	0.65	0.503	0.651
	C380-T4	0.653	0.586	0.658	0.551	0.712
	C380-T6	0.678	0.637	0.625	0.511	0.772
	C393-T3	0.633	0.642	0.555	0.5	0.704
Displacement Mid Rate	v20dt60	0.614	0.572	–	–	0.819
	v20dt30	0.555	0.496	–	–	0.856
	v40dt60	0.65	0.564	–	–	0.826
	v40dt30	0.624	0.49	–	–	0.77
Pressure	Frontal	0.899	0.87	0.922	–	–
	Parietal	0.648	0.673	0.995	–	–
	Occipital 1	0.66	0.696	–	–	–
	Occipital 2	0.591	0.609	–	–	–
Strain	PF	0.626	0.594	0.859	–	–
	C288-T3 C1	0.811	0.679	0.712	0.643	0.808
	C288-T3 C2	0.831	0.818	–	–	–
	C380-T1 C1	0.852	0.707	0.874	0.874	0.782
	C380-T2 C1	0.844	0.795	0.766	0.76	0.721
	C380-T3 C1	0.845	0.868	0.758	0.74	0.9
	C380-T4 C1	0.74	0.91	0.721	0.696	0.961
	C380-T6 C1	0.745	0.909	0.68	0.62	0.893
	C393-T3 C1	0.9	0.839	0.832	0.913	0.802

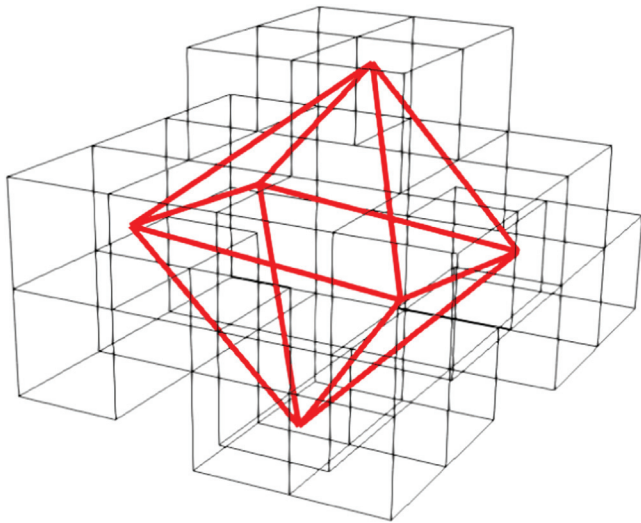


Fig. B.1. Based on the neutral density target (NDT) locations in the experiment, a tetra structure is created in the brain mesh. The elements used for strain computation were then identified as all elements that intersect with or lie inside the tetra structure.

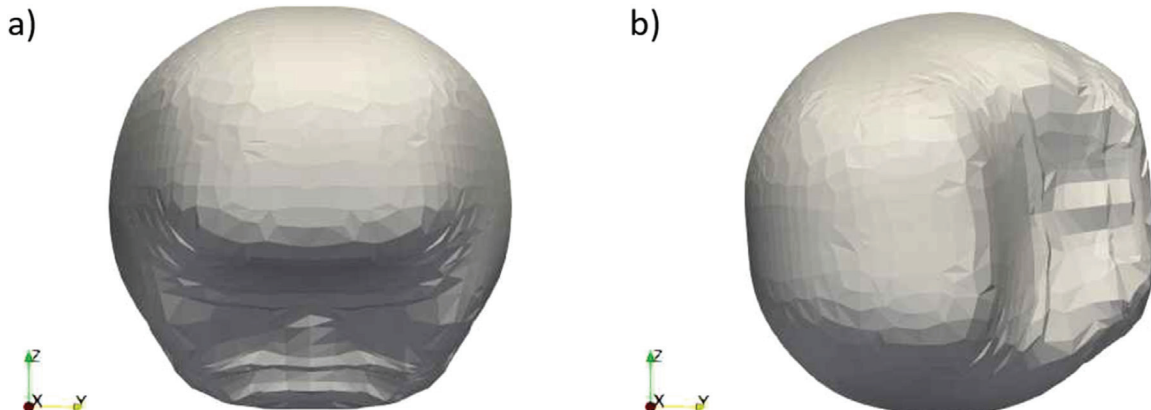


Fig. B.2. Case 1: a) Skull surface before rotation. b) Skull surface after rotation.

sensory cortex, brain stem, cerebellum, frontal lobe, parietal lobe, occipital lobe and temporal lobe.

Case 1: In this case we apply a rotational acceleration about the X axis. Fig. B.2 shows the skull before and after the simulation. Fig. B.3 shows the regional MPS-95 plots as well as the input boundary conditions.

Case 2: Here we apply a combined rotational acceleration about the Y and Z axes. Fig. B.4 shows the skull before and after the simulation. Fig. B.5 shows the regional MPS-95 plots as well as the input boundary conditions.

B.7. Convergence Testing

We have carried out convergence testing using the 95th percentile Maximum Principal Strain metric. The case tested was a relative brain displacement validation case (C380-T4). Fig. B.6 shows the 95th percentile Maximum Principal Strain (MPS-95) for three mesh resolutions: coarse (17,030 elements), fine (105,520 elements) and a newer super-fine mesh (844,160 elements). The trend shows that the coarse and fine brains have not converged. We are additionally running another finer mesh (6,753,280 elements) to test the convergence of the super-fine mesh. However,

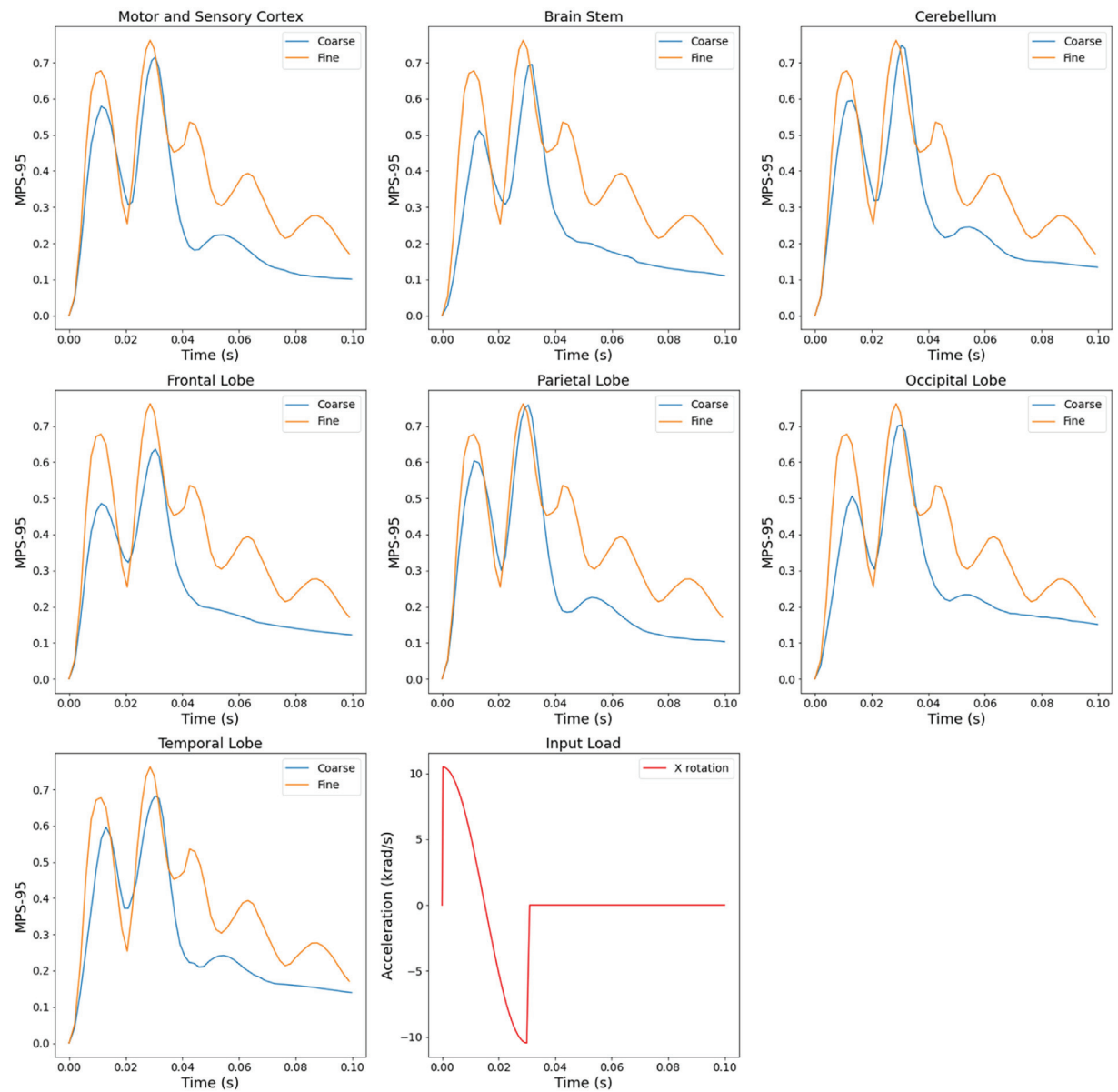


Fig. B.3. Input load curve and MPS-95 curves for the coarse and fine meshes morphed to dimensions of the 50th percentile male head model for benchmarking case 1.

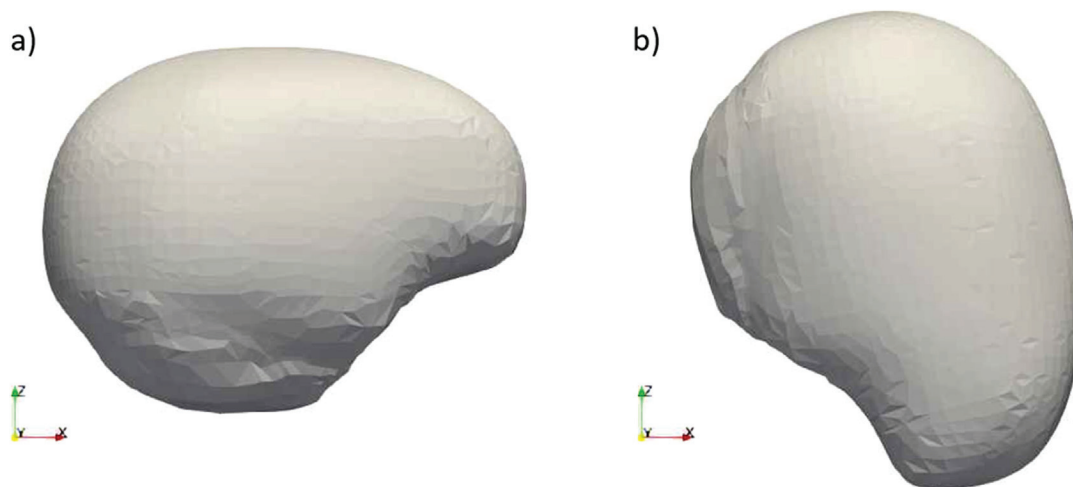


Fig. B.4. Case 2: a) Skull surface before rotation. b) Skull surface after rotation.

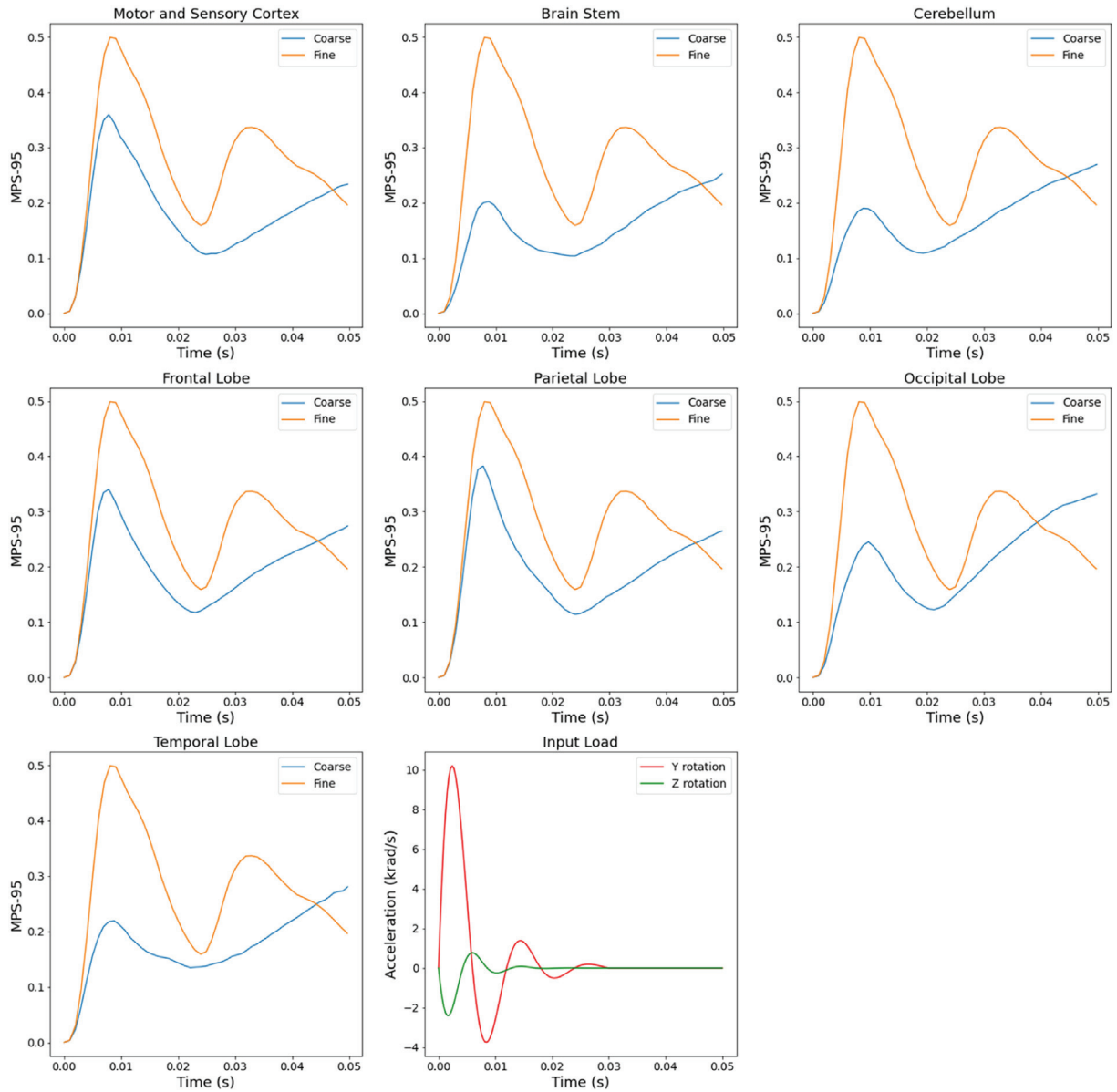


Fig. B.5. Input load curve and MPS-95 curves for the coarse and fine meshes morphed to dimensions of the 50th percentile male head model for benchmarking case 2.

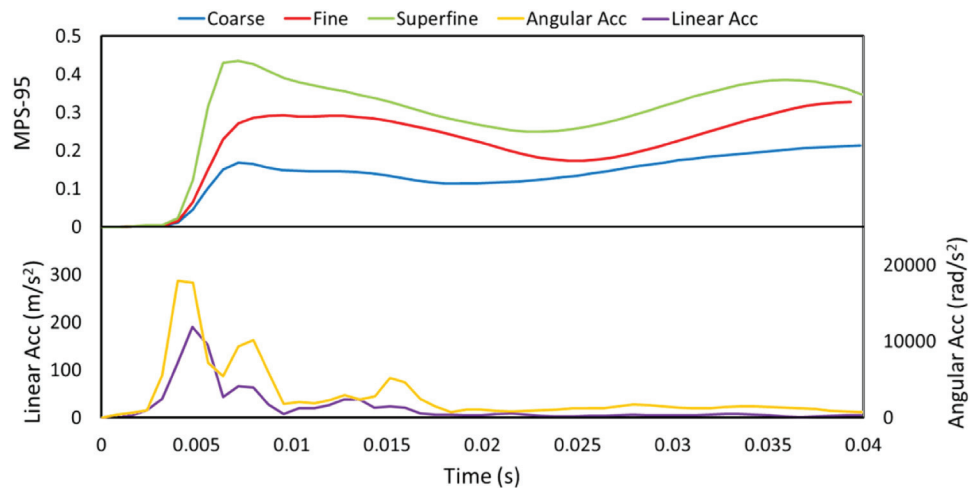


Fig. B.6. MPS-95 results for convergence testing of C380-T4 relative brain-skull displacement validation case.

the simulation takes a significantly long amount of time to run to completion (~300 h using 40 processors). These meshes may be added to the platform as alternatives to the coarse and fine meshes; however the estimate to run one simulation using the super-super-fine mesh is \$1000, which is not feasible for scalable modeling.

The value of the coarse mesh lies in its fast runtime; it is a useful mesh when determining which impact in a long list of impacts results in the largest strains (or other such metrics). These cases can then be rerun using higher resolution meshes to get accurate results. It is also the preferred mesh when testing various features of the platform such as scalability.

It should also be noted that the resolution the coarse brain mesh is over four times the resolution of the original KTH model [30] which has been widely used in literature. By using the same material properties as the KTH model, we were able to get acceptable validation results.

Appendix C. Platform Demonstration:

A test case is used to demonstrate how a user can create a personal mesh, submit one or multiple jobs, and access results.

Create profile and mesh:

The user uploads a picture of their face on the profile page (Fig. C.1), and the platform creates and stores a custom mesh for the user for future use. Where the user does not intend to upload an image or where several cases are being run for which the acceleration data have been collected previously but no associated account/user exists, the platform uses the default meshes created.

Users can see the avatar model created along with the brain by clicking the inspect button. An example is shown in Fig. C.2.

Upload sensor data:

By navigating to the team dashboard, users can submit a new job (see Fig. C.3). They can select the sensor company that collected the impact data, the mesh, and instance type, as shown in Fig. C.4. Our database is then searched for the account to see if a mesh exists (otherwise, the default mesh is used) and if that specific event has already been simulated, in which case users can then choose to remove the simulation.



Fig. C.2. Personalized model and brain mesh created for one user. This model is available for all users and can be found by clicking the inspect button on the custom mesh generation portal.

Alternatively, the user can upload multiple simulations at once; the simulations will be automatically sorted into groups according to the sensor ID, as shown in Fig. C.5. The user can also automatically remove simulations that have already been run.

Monitoring:

Once the job is submitted, the user will be returned to the team dashboard where they can monitor the job status.

The job is highlighted in orange (Fig. C.6) when it is running. Once the job is fully run, the highlight turns green (Fig. C.7).

Visualize results:

Users can then visualize the results for a player/sensor ID by clicking on it from the team dashboard. The user is taken to the player dashboard after clicking (Fig. C.8).

The dashboard displays an interactive image of the brain. The strain events for the user are segregated by the region they occur

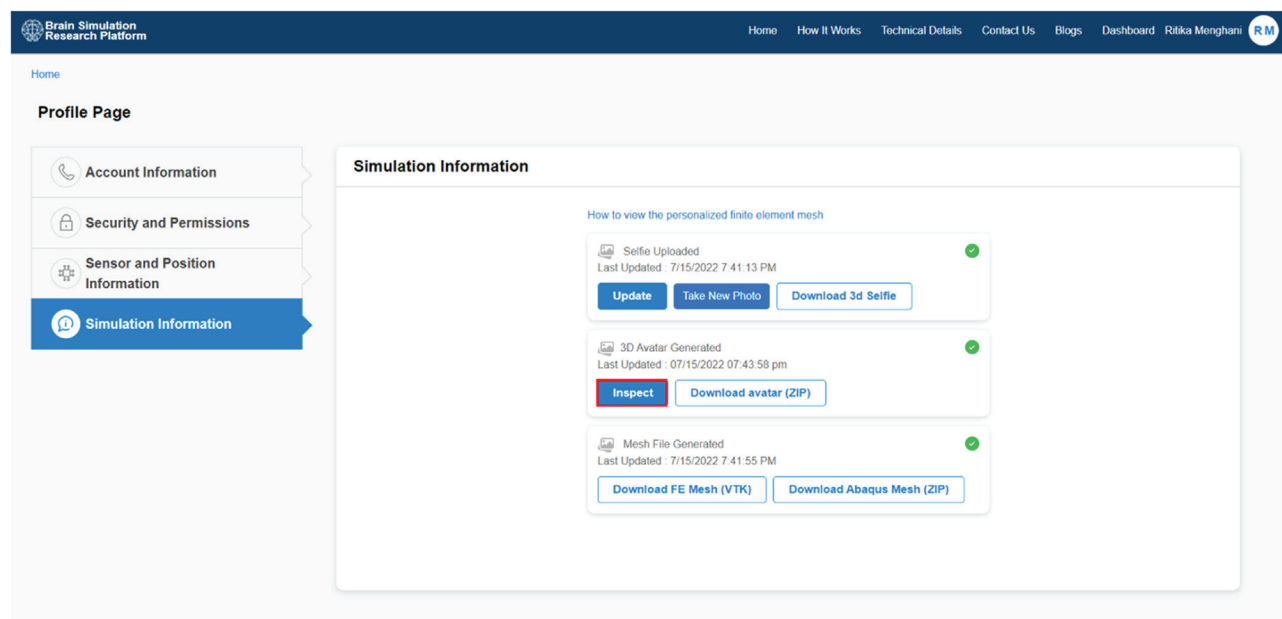


Fig. C.1. Custom mesh creation portal where users can upload a picture of their face and generate a 3D avatar and brain mesh. The meshes are available in VTK and INP file formats.

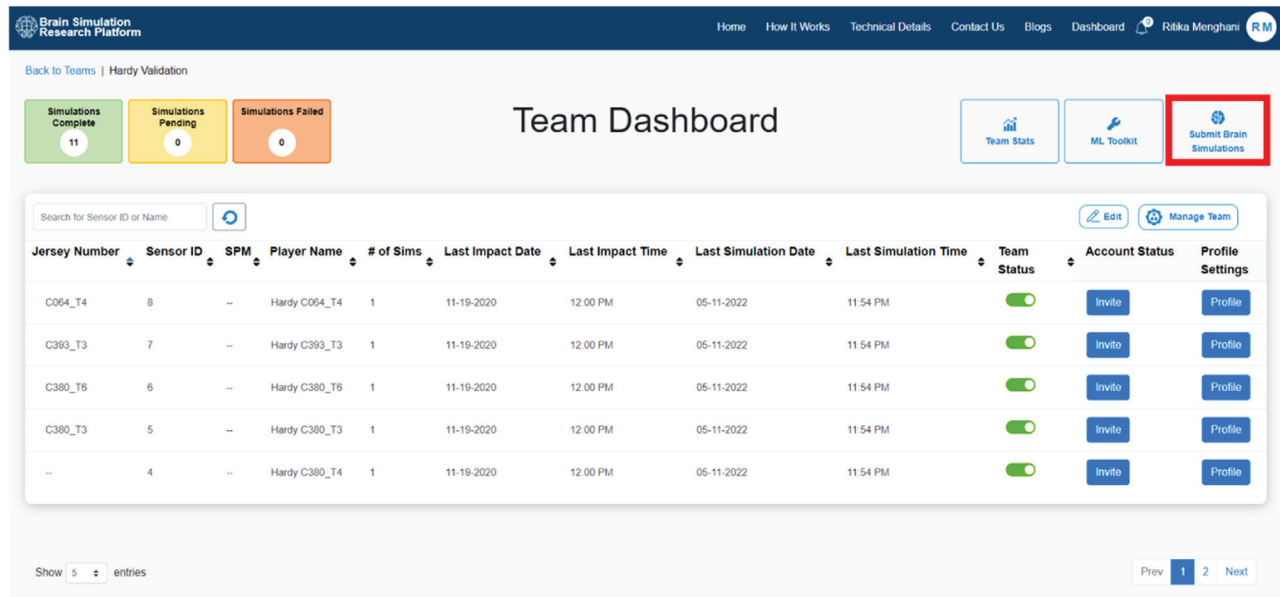


Fig. C.3. Team dashboard that will be visible to any user on the team. All simulations corresponding to the team will be shown here. The red box highlights the button that takes a user to the job submission portal.

Brain Simulation Portal

Upload CSV or JSON files and submit brain simulations right from the web browser.
(Requires Admin Permissions)

Select sensor brand

Upload Sensor Data

A list of different sensor companies can be selected from the dropdown

1 simulation uploaded!

What type of model would you like to use? ☒ Coarse ☐ Fine ☐ Deep Learning

Would you like to use on-demand or spot instances? ☒ On-demand ☐ Spot instances

The user can select the mesh and instance type

Remove any unwanted simulations

Name	Organization	Team	File Name	Account Exist	Profile Image	Overwrite Simulation	Cancel Simulation
Hardy C064_T4	Penn State University	Hardy Validation	input_C064-T4_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	Cancel

The account ID associated with the simulation is compared with existing IDs

Submit

Fig. C.4. Brain submission portal. Users can choose the sensor company, mesh type and instance type for their simulation. They can then upload the acceleration data.

in and are also size and color-coded by their magnitude. By hovering over a region in the major functional region bar graph, the respective area is highlighted, and only strains in that region are shown. Fig. C.9 shows that the strain event occurred in the frontal lobe.

Other plots show the 95th percentile Maximum Principal Strain (MPS-95) with respect to different parameters; these plots are linked such that hovering over one event in one plot will highlight the corresponding event in the other plots (Fig. C.10). A box containing the event ID also appears; this in-

formation can be especially useful when the user has multiple events.

Additional results for this event can be found on the brain simulation details page (Fig. C.12), which can be accessed by selecting the 'view details' option on the player dashboard (Fig. C.11).

The brain simulation details page shown in Fig. C.12 contains several plots and results:

- Two plots show the input accelerations used as boundary conditions.

Brain Simulation Portal

8 simulations uploaded!

What type of model would you like to use? ☒ Coarse ☐ Fine ☐ Deep Learning

Would you like to use on-demand or spot instances? ☒ On-demand ☐ Spot instances

Automatically exclude re-running existing simulations ☒ 0 simulations.

Remove any unwanted simulations

Name	Organization	Team	File Name	Account Exist	Profile Image	Overwrite Simulation	Cancel Simulation
Hardy C393_T3	Penn State University	Hardy Validation	input_C393-T3_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	✖ Cancel
Hardy C064_T4	Penn State University	Hardy Validation	input_C064-T4_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	✖ Cancel
Hardy C288_T3	Penn State University	Hardy Validation	input_C288-T3_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	✖ Cancel
Hardy C380_T1	Penn State University	Hardy Validation	input_C380-T1_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	✖ Cancel
Hardy C380_T2	Penn State University	Hardy Validation	input_C380-T2_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	✖ Cancel
Hardy C380_T3	Penn State University	Hardy Validation	input_C380-T3_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	✖ Cancel
Hardy C380_T4	Penn State University	Hardy Validation	input_C380-T4_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	✖ Cancel
Hardy C380_T6	Penn State University	Hardy Validation	input_C380-T6_Hardyetal2007-ms2.json	Yes Click to expand		Yes Click to expand	✖ Cancel

Submit

Fig. C.5. Job submission portal with multiple jobs uploaded at the same time. Users can choose to remove any simulations they don't want to run by hitting the cancel button under cancel simulation.

- A video shows the skull rigid motion; the user can upload a real-time video of the impact, if any. These videos can be synchronized, and the real and simulated head motion can be observed side-by-side.
- Another video shows the internal brain response to the impact.
- Different injury metrics are also displayed in a plot (see Fig. C.13 for all plots).
- Finally, a plot compares the MPS to a standard curve (T. [66]); this plot can be used to determine if any mild TBI exists for that event. In this case, the MPS value intersects with the curve at almost zero risk, implying that the chance that a mild TBI occurred is low.

Fig. C.13 presents plots showing results from different injury metrics calculated for each simulation. These plots include:

- A ranked MPS plot; this is useful when identifying the 95th percentile maximum principal strain (MPS).

- An image of the brain showing the element that experienced the largest MPS.
- An image of the brain showing the elements that exceed 15% tensile strain (CSDM-15).
- An image of the brain showing the elements that exceed an MPSxSR (product of maximum principal strain and strain rate) of $28s^{-1}$; in this case none of the elements crossed the threshold.
- An image of the brain showing the elements that exceed the 95th percentile maximum principal strain at some point during the simulation (MPS-95).
- An image of the brain showing the elements that exceed 30% tensile strain (CSDM-30); in this case, none of the elements crossed the threshold.

The user can also view the results of the entire team at once on the team analytics page, which can be accessed from the team dashboard (Fig. C.14).

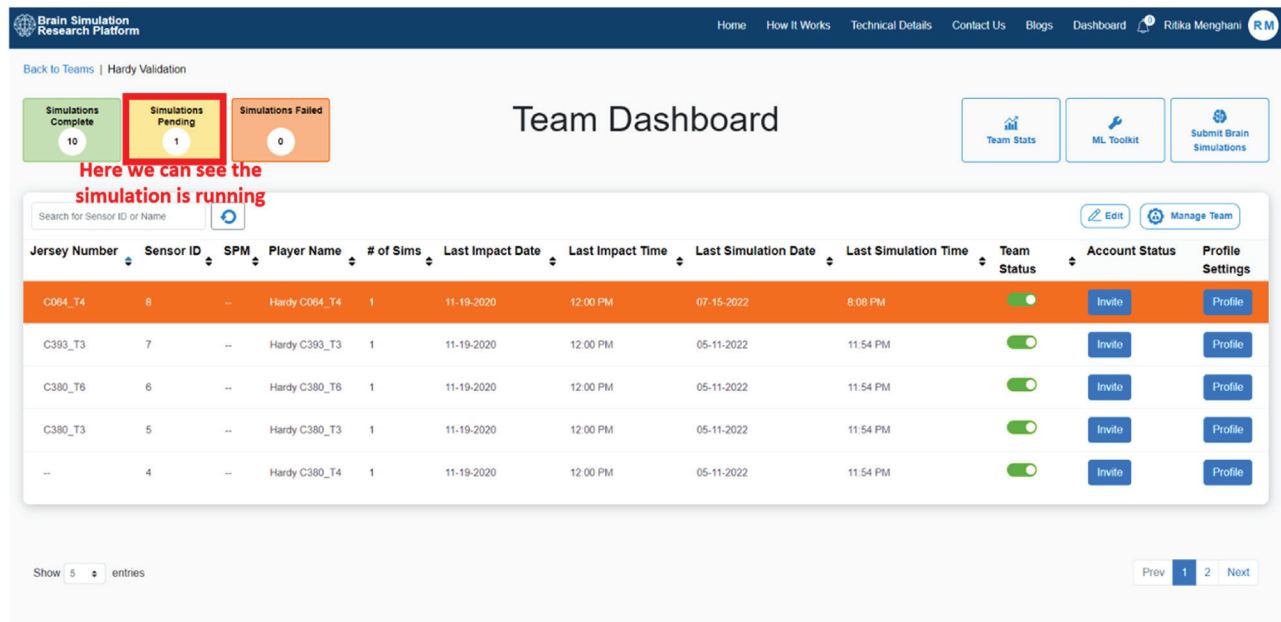


Fig. C.6. Team dashboard after a job is submitted and is running. The red box shows the number of pending simulations. The pending simulations are also highlighted in orange on the dashboard.

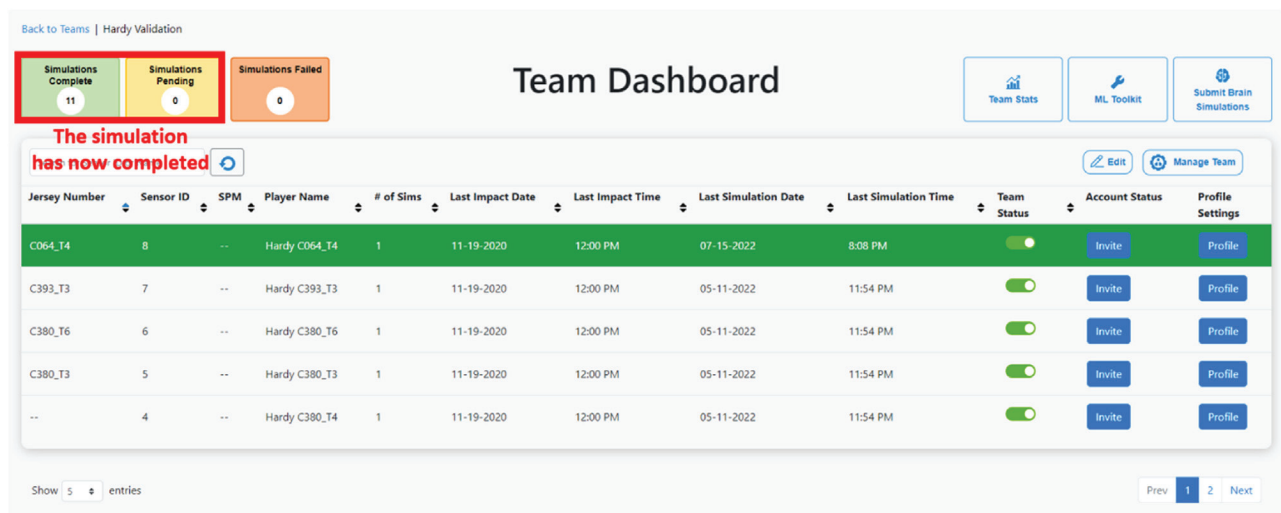


Fig. C.7. Team dashboard after job finishes running. The red box shows the pending simulations as zero and completed simulations as one. The completed simulations are highlighted in green on the dashboard.

The team analytics page (Fig. C.15) appears similar to the player results page but contains additional plots: one that ranks all the accounts according to the 95th percentile MPS and one that shows the time at which the 95th percentile MPS value occurred for each simulation. This plot is linked with the other MPS vs acceleration plots (Fig. C.16).

The user can access the individual results for any event by selecting the event ID when highlighted in the plot (Fig. C.16). This selection leads to the page and results shown in Fig. C.12. Users can also download the results in a clinical style report for a single player from the player dashboard or for the entire team from the team analytics page.

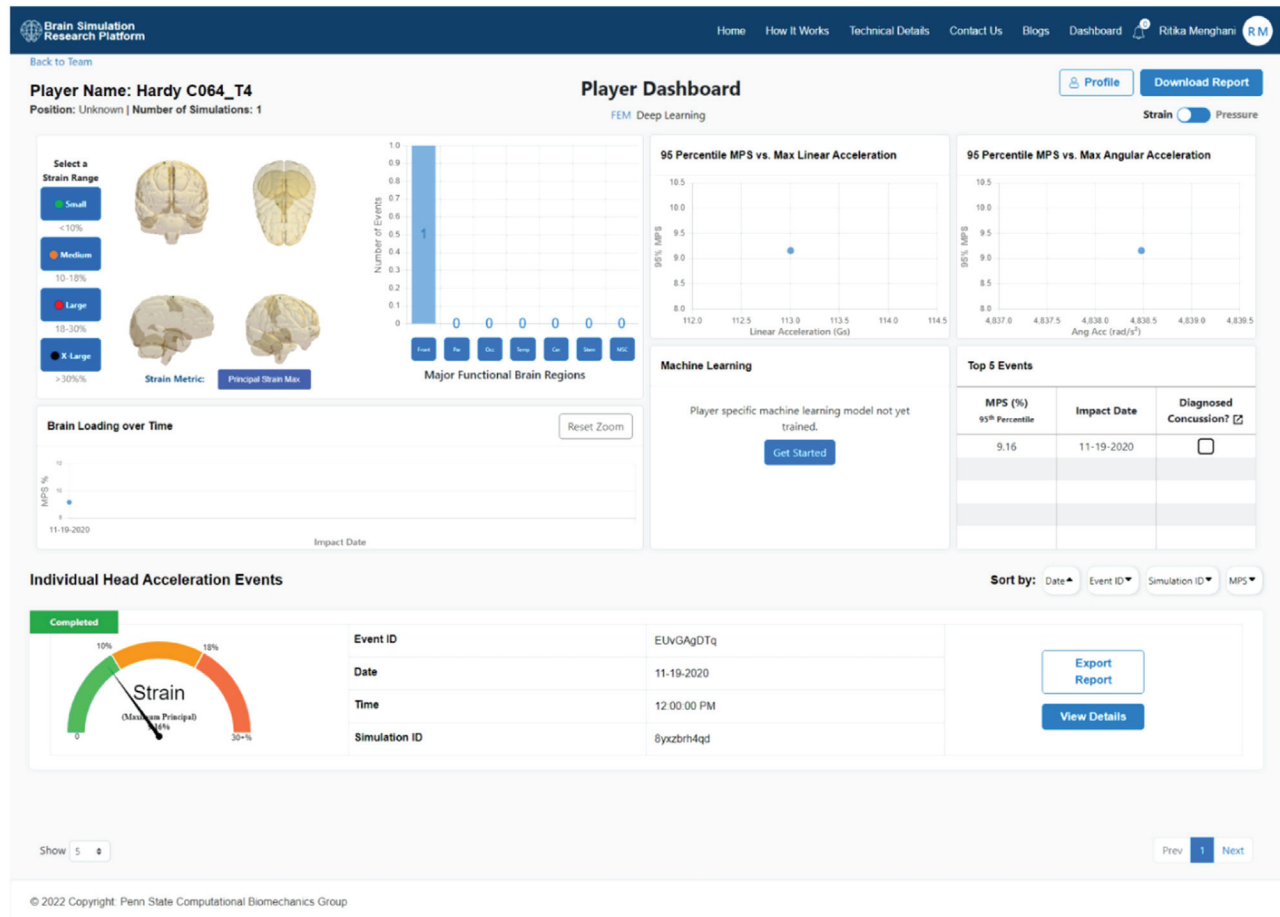


Fig. C.8. The player dashboard shows results of all events experienced by the player. The individual plots have been explained in detail in other figures.

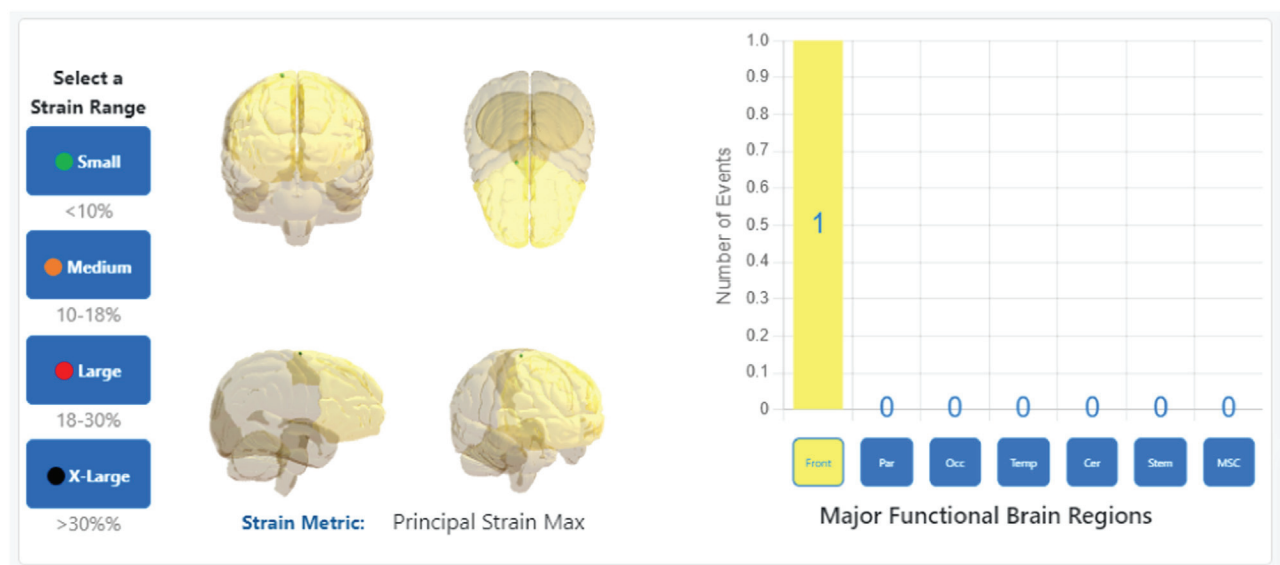


Fig. C.9. The major functional brain regions plot shows the number of events occurring in each of the brain regions. Users can select a specific brain region to visualize strains only in that region. By selecting "front" in the major functional brain regions plot, the frontal lobe and all strains in the frontal lobe are highlighted.

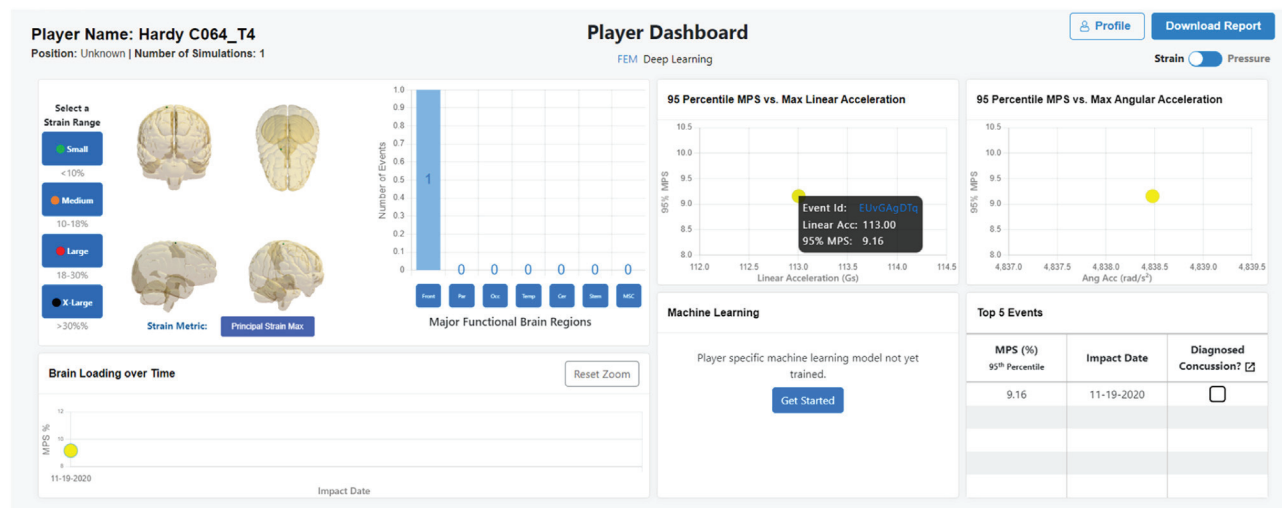


Fig. C.10. The events in the “95th Percentile MPS vs Maximum Linear Acceleration”, the “95th Percentile MPS vs Maximum Angular Acceleration” and the “Brain Loading over Time” plots are linked. By hovering over a plot point in any of these plots, the corresponding plot point highlights in the other plots.

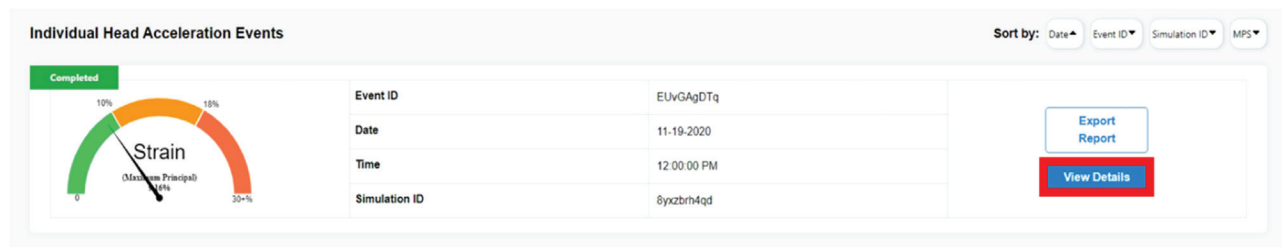


Fig. C.11. The magnitude of the strain event is shown on a scale depicting the level of severity. Further information for this simulation can be found by clicking ‘view details’; this takes users to the “Brain Simulation Results” page.

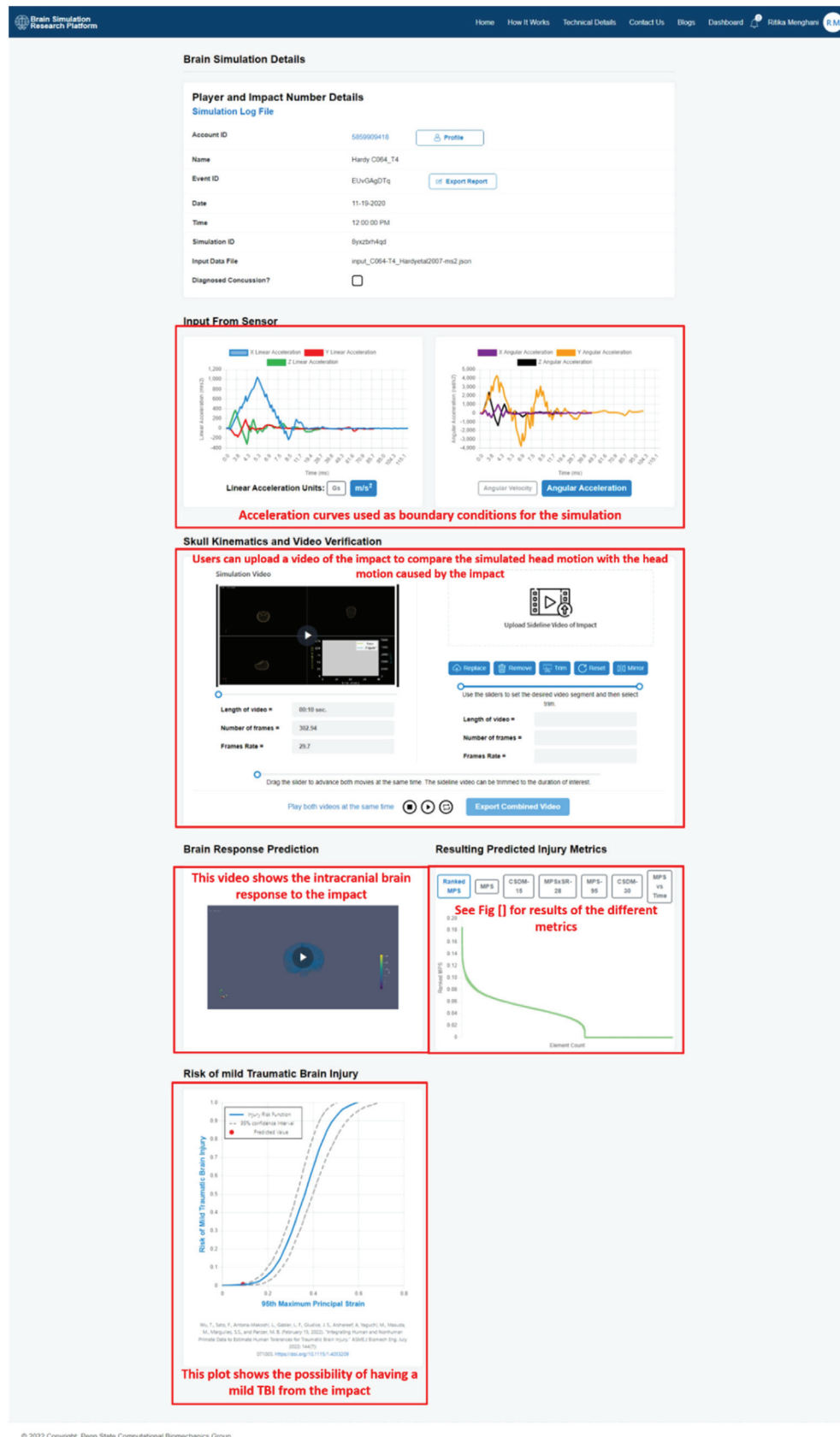


Fig. C.12. All results for any one simulation along with the player information can be found on this page. This includes the input acceleration plots, a video showing skull kinematics that can be synced with a video of the actual impact, a video showing the brain response to the impact, several plots showing various brain injury metrics and one other plot quantifying the risk of a mild TBI from the impact.

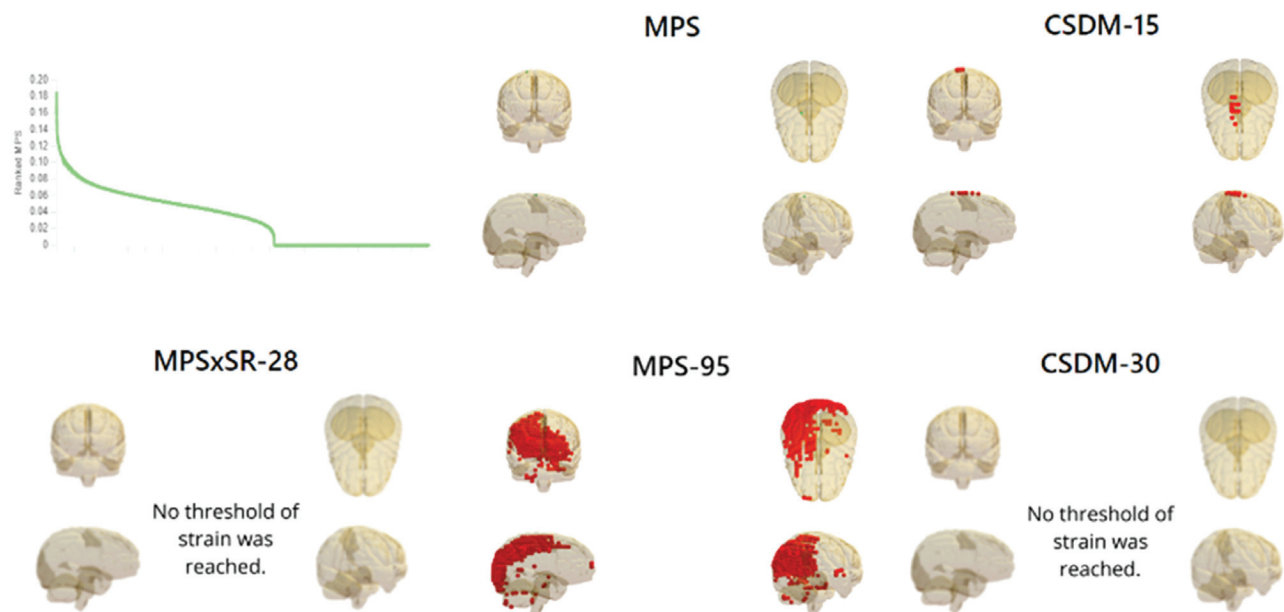


Fig. C.13. Different injury metrics calculated for every simulation. This includes i) a ranked MPS plot; ii) the element that has the largest MPS; iii) elements that exceed 15% tensile strain (CSDM-15); iv) elements that cross $28s^{-1}$ threshold when computing product of MPS and strain rate (MPSxSR-28); v) all elements that exceed 95th percentile MPS (MPS-95); vi) elements that exceed 30% tensile strain (CSDM-30).

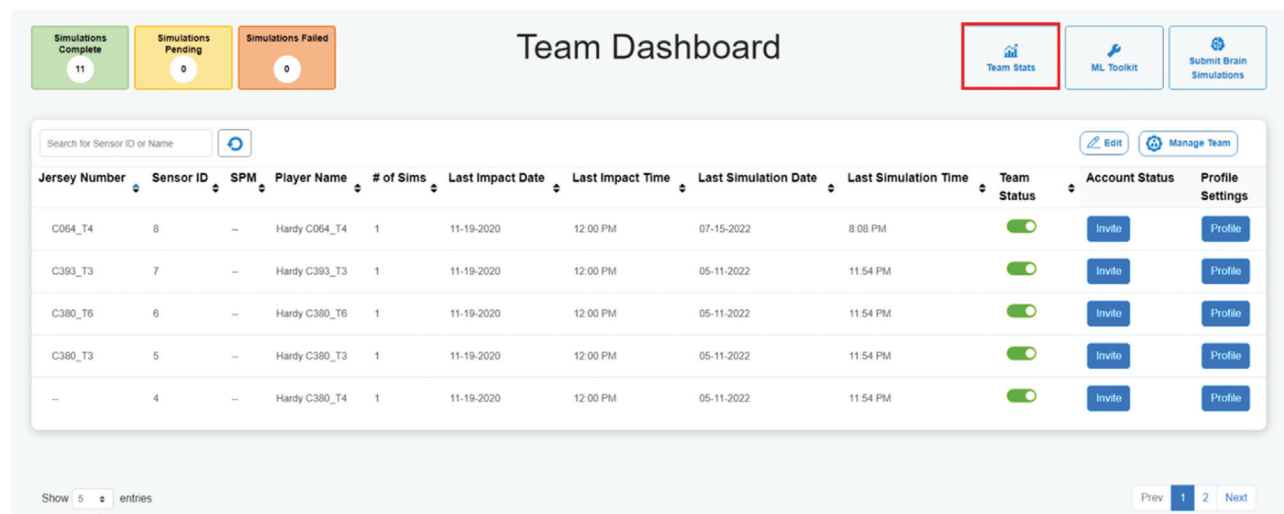


Fig. C.14. Team analytics can be accessed by clicking the "Team Stats" button highlighted with the red box.

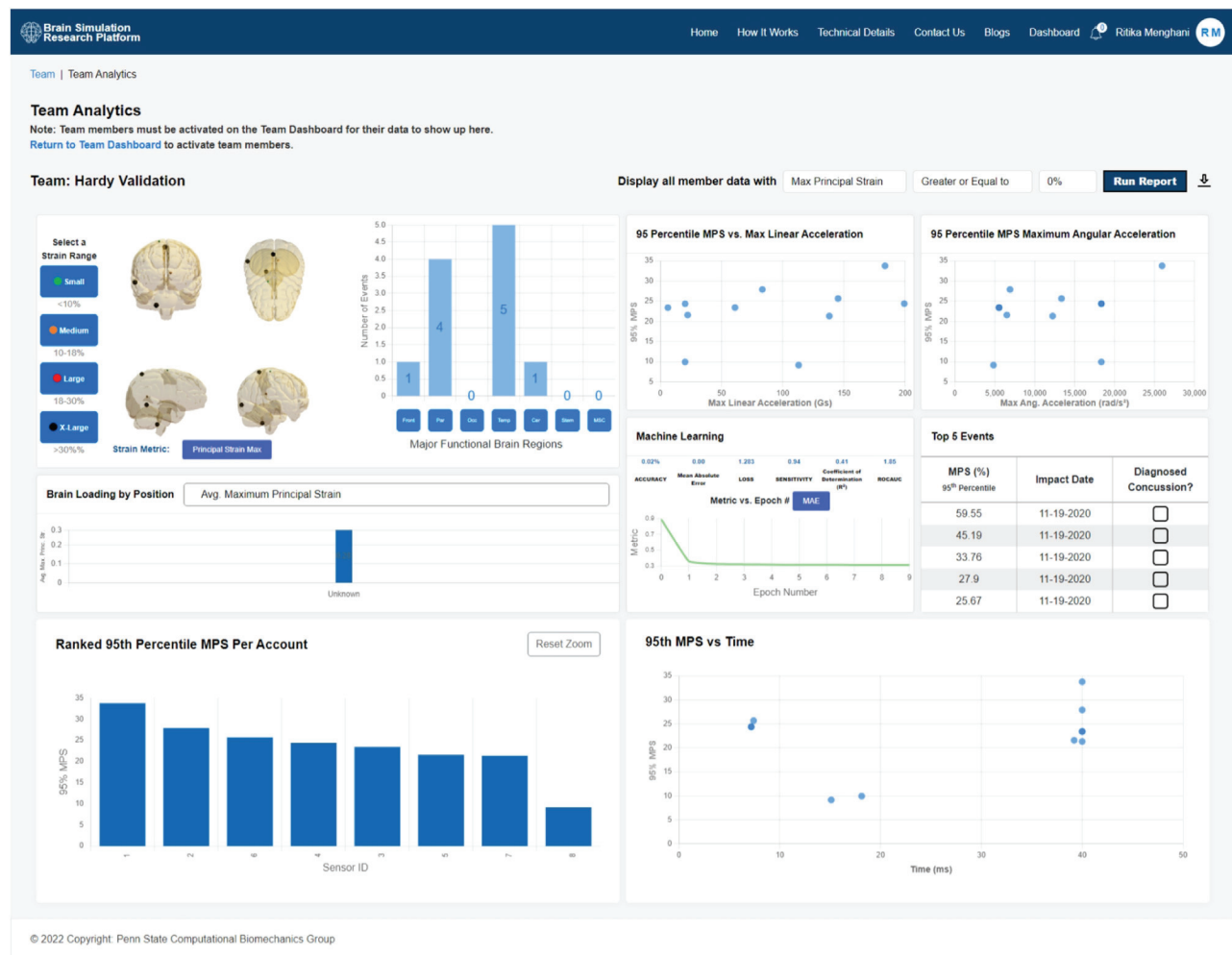


Fig. C.15. Results for all simulations for a team are shown on the team analytics page. These plots are similar to the plots from the player dashboard but also include a ranked 95th percentile MPS vs sensor IDs plot and a 95th percentile MPS vs time plot.

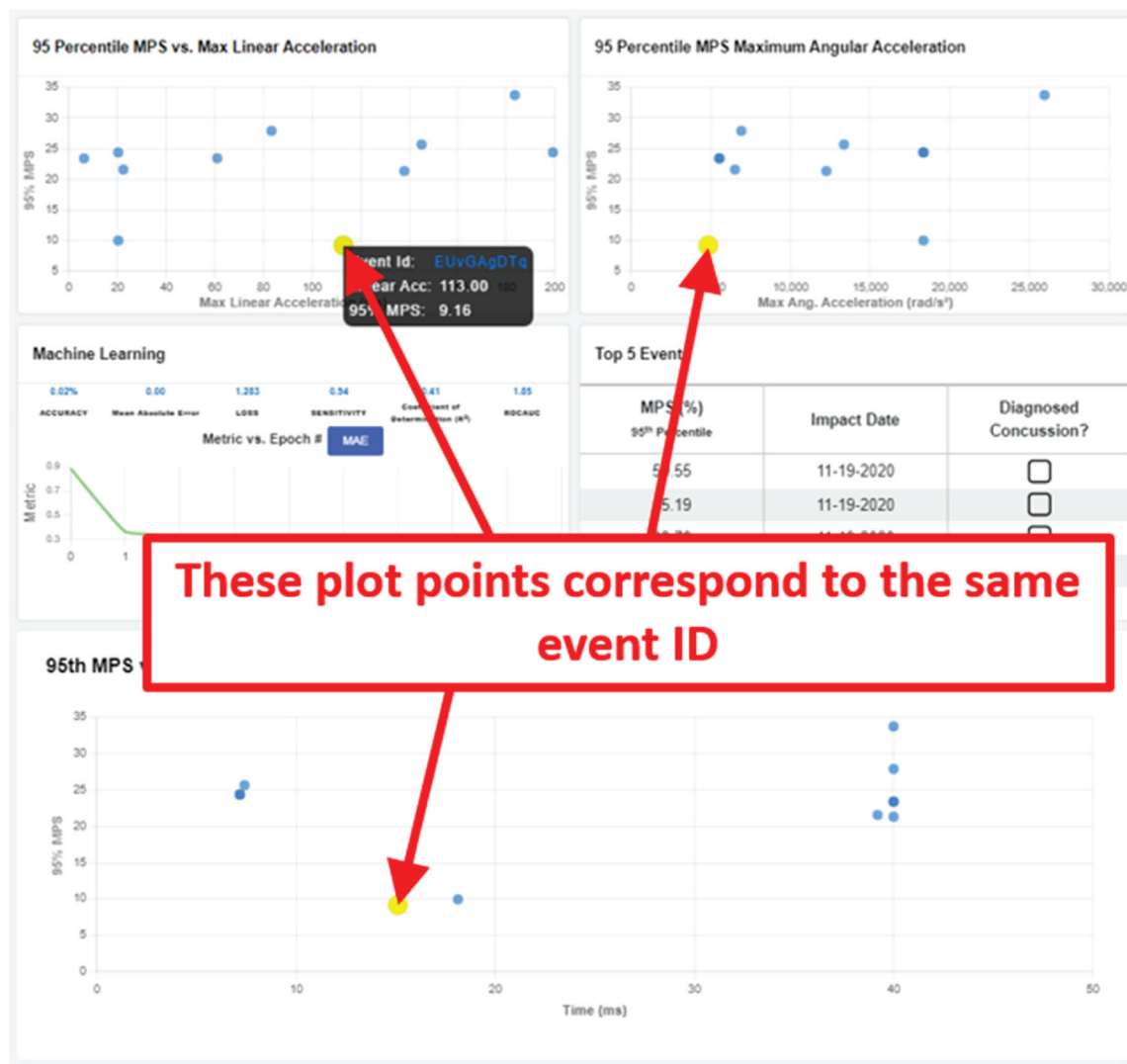


Fig. C.16. The events in the “95th Percentile MPS vs Maximum Linear Acceleration”, the “95th Percentile MPS vs Maximum Angular Acceleration” and the “95th Percentile MPS vs Time” plots are linked. By hovering over a plot point in any of these plots, the corresponding plot point highlights in the other plots. Additionally, a box showing the event details appears by the plot point and by clicking on the event ID, the user will be taken to the “Brain Simulation Results” page for that simulation.

References

- [1] A. Alshareef, J.S. Giudice, J. Forman, R.S. Salzar, M.B. Panzer, A novel method for quantifying human in situ whole brain deformation under rotational loading using sonomicrometry, *J Neurotrauma* 35 (5) (2018) 780–789, doi:[10.1089/neu.2017.5362](https://doi.org/10.1089/neu.2017.5362).
- [2] A. Alshareef, A.K. Knutsen, C.L. Johnson, A. Carass, K. Upadhyay, P.V. Bayly, D.L. Pham, J.L. Prince, K.T. Ramesh, Integrating material properties from magnetic resonance elastography into subject-specific computational models for the human brain, *Brain Multiphysics* 2 (2021) 100038, doi:[10.1016/j.brain.2021.100038](https://doi.org/10.1016/j.brain.2021.100038).
- [3] A. Alshareef, T. Wu, J.S. Giudice, M.B. Panzer, Toward subject-specific evaluation: methods of evaluating finite element brain models using experimental high-rate rotational brain motion, *Biomech Model Mechanobiol* 20 (6) (2021) 2301–2317, doi:[10.1007/s10237-021-01508-7](https://doi.org/10.1007/s10237-021-01508-7).
- [4] A.J. Bartsch, M.M. McCrea, D.S. Hedin, P.L. Gibson, V.J. Miele, E.C. Benzell, J.L. Alberts, S. Samorezov, A. Shah, B.S. Stemper, Laboratory and on-field data collected by a head impact monitoring mouthguard, in: 2019 41st Annu Int Conf IEEE Eng Med Biol Soc IEEE Eng Med Biol Soc Annu Int Conf., 2019, IEEE, 2019, pp. 2068–2072, doi:[10.1109/EMBC.2019.8856907](https://doi.org/10.1109/EMBC.2019.8856907).
- [5] T. Belytschko, W.K. Liu, B. Moran, K. Elkhodary, *Nonlinear Finite Elements For Continua and Structures*, John Wiley & Sons, New York, United Kingdom, 2014 Incorporated.
- [6] H.S. Chan, *Mathematical Model for Closed Head Impact*, *SAE Trans* 83 (1974) 3814–3825.
- [7] Y. Chen, T.J. O’Shaughnessy, G.H. Kamimori, D.M. Horner, M.J. Egnoto, A. Bagchi, Role of interfacial conditions on blast overpressure propagation into the brain, *Front Neurol* 11 (2020) 323, doi:[10.3389/fneur.2020.00323](https://doi.org/10.3389/fneur.2020.00323).
- [8] K. Christianson, V. Owusu, M. Taylor, D. Hopfe, P. Pavilonis, N.G. Murray, A-09 Challenges in measuring repetitive head impacts, cognition, and eye movements in high school football players, *Arch Clin Neuropsychol* 37 (5) (2022) 1050, doi:[10.1093/arclin/acac32.09](https://doi.org/10.1093/arclin/acac32.09).
- [9] G. Fanti, A. Spinazzè, F. Borghi, S. Rovelli, D. Campagnolo, M. Keller, A. Borghi, A. Cattaneo, E. Cauda, D.M. Cavallo, Evolution and applications of recent sensing technology for occupational risk assessment: a rapid review of the literature, *Sensors* 22 (13) (2022) 4841, doi:[10.3390/s22134841](https://doi.org/10.3390/s22134841).
- [10] F.A. Fernandes, D. Tchepel, R.J.A. de Sousa, M. Ptak, Development and validation of a new finite element human head model: yet another head model (YEAHM), *Eng Comput* 35 (1) (2018) 477–496, doi:[10.1108/EC-09-2016-0321](https://doi.org/10.1108/EC-09-2016-0321).
- [11] D.P. Flanagan, T. Belytschko, A uniform strain hexahedron and quadrilateral with orthogonal hourglass control, *Int J Numer Methods Eng* 17 (5) (1981) 679–706, doi:[10.1002/nme.1620170504](https://doi.org/10.1002/nme.1620170504).
- [12] L.F. Gabler, S.H. Huddleston, N.Z. Dau, D.J. Lessley, K.B. Arbogast, X. Thompson, J.E. Resch, J.R. Crandall, On-field performance of an instrumented mouthguard for detecting head impacts in american football, *Ann Biomed Eng* 48 (11) (2020) 2599–2612, doi:[10.1007/s10439-020-02654-2](https://doi.org/10.1007/s10439-020-02654-2).
- [13] S. Ganpule, N.P. Daphalapurkar, K.T. Ramesh, A.K. Knutsen, D.L. Pham, P.V. Bayly, J.L. Prince, A three-dimensional computational human head model that captures live human brain dynamics, *J Neurotrauma* 34 (13) (2017) 2154–2166, doi:[10.1089/neu.2016.4744](https://doi.org/10.1089/neu.2016.4744).
- [14] M. Ghajari, P.J. Hellyer, D.J. Sharp, Computational modelling of traumatic brain injury predicts the location of chronic traumatic encephalopathy pathology, *Brain* 140 (2) (2017) 333–343, doi:[10.1093/brain/aww317](https://doi.org/10.1093/brain/aww317).
- [15] K. Ghazi, S. Wu, W. Zhao, S. Ji, Instantaneous whole-brain strain estimation in dynamic head impact, *J Neurotrauma* 38 (8) (2021) 1023–1035, doi:[10.1089/neu.2020.7281](https://doi.org/10.1089/neu.2020.7281).

- [16] C. Giordano, S. Kleiven, Evaluation of axonal strain as a predictor for mild traumatic brain injuries using finite element modeling, *Stapp Car Crash J* 58 (2014) 29–62.
- [17] C. Giordano, S. Kleiven, Development of an unbiased validation protocol to assess the biofidelity of finite element head models used in prediction of traumatic brain injury, *Stapp Car Crash J* 60 (2016) 363–471, doi:[10.4271/2016-22-0013](https://doi.org/10.4271/2016-22-0013).
- [18] J.S. Giudice, A. Alshareef, T. Wu, C.A. Gancayco, K.A. Reynier, N.J. Tustison, T.J. Druzgal, M.B. Panzer, An image registration-based morphing technique for generating subject-specific brain finite element models, *Ann Biomed Eng* 48 (10) (2020) 2412–2424, doi:[10.1007/s10439-020-02584-z](https://doi.org/10.1007/s10439-020-02584-z).
- [19] D.G. Greybe, C.M. Jones, M.R. Brown, E.M.P. Williams, Comparison of head impact measurements via an instrumented mouthguard and an anthropometric testing device, *Sports Eng* 23 (1) (2020) 12, doi:[10.1007/s12283-020-00324-z](https://doi.org/10.1007/s12283-020-00324-z).
- [20] M. Hajiaghameh, T. Wu, M.B. Panzer, S.S. Margulies, Embedded axonal fiber tracts improve finite element model predictions of traumatic brain injury, *Biomech Model Mechanobiol* 19 (3) (2020) 1109–1130, doi:[10.1007/s10237-019-01273-8](https://doi.org/10.1007/s10237-019-01273-8).
- [21] W.N. Hardy, *Response of the Human Cadaver Head to Impact*, Wayne State University, Detroit, Michigan, 2007.
- [22] W.N. Hardy, M.J. Mason, C.D. Foster, C.S. Shah, J.M. Kopacz, K.H. Yang, A.I. King, J. Bishop, M. Bey, W. Anderst, S. Tashman, A study of the response of the human cadaver head to impact, *Stapp Car Crash J* 51 (2007) 17–80, doi:[10.4271/2007-22-0002](https://doi.org/10.4271/2007-22-0002).
- [23] D.S. Hedin, P.L. Gibson, A.J. Bartsch, S. Samorezov, Development of a head impact monitoring "Intelligent Mouthguard", in: 2016 38th Annu Int Conf IEEE Eng Med Biol Soc EMBC, IEEE, 2016, pp. 2007–2009, doi:[10.1109/EMBC.2016.7591119](https://doi.org/10.1109/EMBC.2016.7591119).
- [24] A.E. Jansen, M. McGrath, S. Samorezov, J. Johnston, A. Bartsch, J. Alberts, Characterizing head impact exposure in men and women during boxing and mixed martial arts, *Orthop J Sports Med* 9 (12) (2021) 23259671211059816.
- [25] S. Ji, M. Ghajari, H. Mao, R.H. Kraft, M. Hajiaghameh, M.B. Panzer, R. Willinger, M.D. Gilchrist, S. Kleiven, J.D. Stitzel, Use of brain biomechanical models for monitoring impact exposure in contact sports, *Ann Biomed Eng* (2022), doi:[10.1007/s10439-022-02999-w](https://doi.org/10.1007/s10439-022-02999-w).
- [26] S. Ji, W. Zhao, J.C. Ford, J.G. Beckwith, R.P. Bolander, R.M. Greenwald, L.A. Flashman, K.D. Paulsen, T.W. McAllister, Group-wise evaluation and comparison of white matter fiber strain and maximum principal strain in sports-related concussion, *J Neurotrauma* 32 (7) (2015) 441–454, doi:[10.1089/neu.2013.3268](https://doi.org/10.1089/neu.2013.3268).
- [27] B. Jones, J. Tooby, D. Weaving, K. Till, C. Owen, M. Begonia, K.A. Stokes, S. Rowson, G. Phillips, S. Hendricks, R.H. Kraft, Ready for impact? A validity and feasibility study of instrumented mouthguards (iMGs), *Br J Sports Med* (2022) bjsports-2022-105523, doi:[10.1136/bjsports-2022-105523](https://doi.org/10.1136/bjsports-2022-105523).
- [28] M. Kaliske, H. Rothert, Formulation and implementation of three-dimensional viscoelasticity at small and finite strains, *Comput Mech* 19 (3) (1997) 228–239, doi:[10.1007/s004660050171](https://doi.org/10.1007/s004660050171).
- [29] E.E. Kieffer, C. Vaillancourt, P.G. Brolinson, S. Rowson, Using in-mouth sensors to measure head kinematics in rugby, *IRCOBI Conf* 13 (2020) 846–858 [place unknown].
- [30] S. Kleiven, Predictors for traumatic brain injuries evaluated through accident reconstructions, *Stapp Car Crash J* 51 (2007) 81–114, doi:[10.4271/2007-22-0003](https://doi.org/10.4271/2007-22-0003).
- [31] A.K. Knutsen, E. Magrath, J.E. McEntee, F. Xing, J.L. Prince, P.V. Bayly, J.A. Butman, D.L. Pham, Improved measurement of brain deformation during mild head acceleration using a novel tagged MRI sequence, *J Biomech* 47 (14) (2014) 3475–3481, doi:[10.1016/j.jbiomech.2014.09.010](https://doi.org/10.1016/j.jbiomech.2014.09.010).
- [32] C. Kuo, D. Patton, T. Rooks, G. Tierney, A. McIntosh, R. Lynall, A. Esquivel, R. Daniel, T. Kaminski, J. Mihalik, et al., On-Field Deployment and Validation for Wearable Devices, *Ann Biomed Eng* 50 (11) (2022) 1372–1388, doi:[10.1007/s10439-022-03001-3](https://doi.org/10.1007/s10439-022-03001-3).
- [33] E. Le Flao, P. Hume, D. King, Head impact monitoring: what new methodologies could do for concussion biomechanics, *ISBS Proc Arch* 36 (1) (2018) 1041.
- [34] X. Li, Z. Zhou, S. Kleiven, An anatomically detailed and personalizable head injury model: significance of brain and white matter tract morphological variability on strain, *Biomech Model Mechanobiol* 20 (2) (2021) 403–431, doi:[10.1007/s10237-020-01391-8](https://doi.org/10.1007/s10237-020-01391-8).
- [35] Z. Li, X. Han, H. Ge, C. Ma, A semi-automatic method of generating subject-specific pediatric head finite element models for impact dynamic responses to head injury, *J Mech Behav Biomed Mater* 60 (2016) 557–567, doi:[10.1016/j.jmbbm.2016.03.021](https://doi.org/10.1016/j.jmbbm.2016.03.021).
- [36] Z. Li, J. Hu, J. Zhang, Comparison of different radial basis functions in developing subject-specific infant head finite element models for injury biomechanics study, in: ASME 2012 Summer Bioeng Conf Parts B, American Society of Mechanical Engineers, Fajardo, Puerto Rico, USA, 2012, pp. 49–50, doi:[10.1115/SBC2012-80162](https://doi.org/10.1115/SBC2012-80162).
- [37] J. Liu, J. (Judy) Jin, J.T. Eckner, S. Ji, J. Hu, Influence of morphological variation on brain impact responses among youth and young adults, *J Biomech* 135 (2022) 111036, doi:[10.1016/j.jbiomech.2022.111036](https://doi.org/10.1016/j.jbiomech.2022.111036).
- [38] Y. Liu, A.G. Domel, N.J. Cecchi, E. Rice, A.A. Callan, S.J. Raymond, Z. Zhou, X. Zhan, Y. Li, M.M. Zeineh, et al., Time window of head impact kinematics measurement for calculation of brain strain and strain rate in American football, *Ann Biomed Eng* 49 (10) (2021) 2791–2804, doi:[10.1007/s10439-021-02821-z](https://doi.org/10.1007/s10439-021-02821-z).
- [39] LS-DYNA Theory Manual - March 2006:680.
- [40] R. Mannix, K. Kawata, J.J. Bazarian, Defining an approach to monitoring brain health in individuals exposed to repetitive head impacts: lessons learned from radiation safety, *J Neurotrauma* 39 (13–14) (2022) 897–901, doi:[10.1089/neu.2022.29130.rm](https://doi.org/10.1089/neu.2022.29130.rm).
- [41] H. Mao, L. Zhang, B. Jiang, V.V. Genthikatti, X. Jin, F. Zhu, R. Makwana, A. Gill, G. Jandir, A. Singh, K.H. Yang, Development of a finite element human head model partially validated with thirty five experimental cases, *J Biomech Eng* 135 (11) (2013) 111002, doi:[10.1115/1.4025101](https://doi.org/10.1115/1.4025101).
- [42] K.-A. Mardal, M.E. Rognes, T.B. Thompson, L.M. Valnes, *Mathematical Modeling of the Human Brain: From magnetic Resonance Images to Finite Element Simulation*, Springer Nature, 2022 [place unknown].
- [43] T.W. McAllister, J.C. Ford, S. Ji, J.G. Beckwith, L.A. Flashman, K. Paulsen, R.M. Greenwald, Maximum principal strain and strain rate associated with concussion diagnosis correlates with changes in corpus callosum white matter indices, *Ann Biomed Eng* 40 (1) (2012) 127–140, doi:[10.1007/s10439-011-0402-6](https://doi.org/10.1007/s10439-011-0402-6).
- [44] L.E. Miller, J.E. Urban, J.D. Stitzel, Development and validation of an atlas-based finite element brain model, *Biomech Model Mechanobiol* 15 (5) (2016) 1201–1214, doi:[10.1007/s10237-015-0754-1](https://doi.org/10.1007/s10237-015-0754-1).
- [45] A. Misistia, M. Skotak, A. Cardenas, E. Alay, N. Chandra, G.H. Kamimori, Sensor orientation and other factors which increase the blast overpressure reporting errors, *PLoS ONE* 15 (10) (2020) e0240262, doi:[10.1371/journal.pone.0240262](https://doi.org/10.1371/journal.pone.0240262).
- [46] A. Montanino, X. Li, Z. Zhou, M. Zeineh, D. Camarillo, S. Kleiven, Subject-specific multiscale analysis of concussion: from macroscopic loads to molecular-level damage, *Brain Multiphysics* 2 (2021) 100027, doi:[10.1016/j.brain.2021.100027](https://doi.org/10.1016/j.brain.2021.100027).
- [47] A.M. Nahum, R. Smith, C.C. Ward, *Intracranial Pressure Dynamics During Head Impact*, SAE Technical Paper, 1977 [place unknown].
- [48] D.A. Patton, C.M. Huber, D. Fedonni, S.S. Margulies, C.L. Master, K.B. Arbogast, Quantifying head impact exposure, mechanisms and kinematics using instrumented mouthguards in female high school lacrosse, *Res Sports Med Print* (2022) 1–15, doi:[10.1080/15438627.2022.2042294](https://doi.org/10.1080/15438627.2022.2042294).
- [49] S.J. Raymond, N.J. Cecchi, H.V. Alizadeh, A.A. Callan, E. Rice, Y. Liu, Z. Zhou, M. Zeineh, D.B. Camarillo, Physics-informed machine learning improves detection of head impacts, *Ann Biomed Eng* (2022), doi:[10.1007/s10439-022-02911-6](https://doi.org/10.1007/s10439-022-02911-6).
- [50] A. Rezaei, L.C. Wu, Automated soccer head impact exposure tracking using video and deep learning, *Sci Rep* 12 (1) (2022) 1–12.
- [51] J.S. Ruan, T.B. Khalil, A.I. King, *Finite Element Modeling of Direct Head Impact*, SAE Technical Paper, 1993 [place unknown].
- [52] M. Schirner, L. Domide, D. Perdiki, P. Triebkorn, L. Stefanovski, R. Pai, P. Prodan, B. Valean, J. Palmer, C. Langford, et al., Brain simulation as a cloud service: the Virtual Brain on EBRAINS, *Neuroimage* 251 (2022) 118973, doi:[10.1016/j.neuroimage.2022.118973](https://doi.org/10.1016/j.neuroimage.2022.118973).
- [53] H. Takao, D. Watanabe, S. Tani, H. Ohashi, T. Ishibashi, K. Takeshita, S. Murakami, T. Nishimoto, K. Yuge, K. Karagiozov, et al., Use of a simulation model to investigate the mechanisms of sports-related head injuries, *Neurol Med Chir (Tokyo)* 62 (1) (2022) 13–18, doi:[10.2176/nmc.2021-0149](https://doi.org/10.2176/nmc.2021-0149).
- [54] E.G. Takhounts, S.A. Ridella, V. Hasija, R.E. Tannous, J.Q. Campbell, D. Malone, K. Danelson, J. Stitzel, S. Rowson, S. Duma, Investigation of traumatic brain injuries using the next generation of Simulated Injury Monitor (SIMon) finite element head model, *Stapp Car Crash J* 52 (2008) 1, doi:[10.4271/2008-22-0001](https://doi.org/10.4271/2008-22-0001).
- [55] X.G. Tan, A. Bagchi, Computational analysis for validation of blast induced traumatic brain injury and protection of combat helmet, *ASME 2018 International Mechanical Engineering Congress and Exposition: American Society of Mechanical Engineers Digital Collection*, 2018, doi:[10.1115/IMECE2018-87689](https://doi.org/10.1115/IMECE2018-87689).
- [56] P.A. Taylor, C.C. Ford, Simulation of blast-induced early-time intracranial wave physics leading to traumatic brain injury, *J Biomech Eng* 131 (6) (2009) 061007, doi:[10.1115/1.3118765](https://doi.org/10.1115/1.3118765).
- [57] M. Tezzele, N. Demo, A. Mola, G. Rozza, PyGeM: python Geometrical Morphing, *Softw Impacts*. 7 (2021) 100047, doi:[10.1016/j.simpa.2020.100047](https://doi.org/10.1016/j.simpa.2020.100047).
- [58] Thunert C. CORA Release 3.6 User's Manual:68.
- [59] A. Trotta, J.M. Clark, A. McGoldrick, M.D. Gilchrist, A.N. Annaidh, Biofidelic finite element modelling of brain trauma: importance of the scalp in simulating head impact, *Int J Mech Sci* 173 (2020) 105448, doi:[10.1016/j.jimecs.2020.105448](https://doi.org/10.1016/j.jimecs.2020.105448).
- [60] T. Tunthanathip, T. Oearsakul, Application of machine learning to predict the outcome of pediatric traumatic brain injury, *Chin J Traumatol* 24 (6) (2021) 350–355, doi:[10.1016/j.cjtee.2021.06.003](https://doi.org/10.1016/j.cjtee.2021.06.003).
- [61] N.A. Vavalle, S.L. Schoell, A.A. Weaver, J.D. Stitzel, F.S. Gayzik, Application of radial basis function methods in the development of a 95th percentile male seated FEA model, *Stapp Car Crash J* 58 (2014) 361, doi:[10.4271/2014-22-0013](https://doi.org/10.4271/2014-22-0013).
- [62] C.C. Ward, R.B. Thompson, The development of a detailed finite element brain model, *SAE Trans.* (1975) 3238–3252, doi:[10.4271/751163](https://doi.org/10.4271/751163).
- [63] A. Wei, J. Wang, J. Liu, M.L.H. Jones, J. Hu, A parametric head geometry model accounting for variation among adolescent and young adult populations, *Comput Methods Programs Biomed* 220 (2022) 106805, doi:[10.1016/j.cmpb.2022.106805](https://doi.org/10.1016/j.cmpb.2022.106805).
- [64] R.M. Wright, K.T. Ramesh, An axonal strain injury criterion for traumatic brain injury, *Biomech Model Mechanobiol* 11 (1–2) (2012) 245–260, doi:[10.1007/s10237-011-0307-1](https://doi.org/10.1007/s10237-011-0307-1).
- [65] S. Wu, W. Zhao, S. Barbat, J. Ruan, S. Ji, Instantaneous brain strain estimation for automotive head impacts via deep learning, *Stapp Car Crash J* 65 (2021) 139–162, doi:[10.4271/2021-22-0006](https://doi.org/10.4271/2021-22-0006).
- [66] S. Wu, W. Zhao, S. Ji, Real-time dynamic simulation for highly accurate spatiotemporal brain deformation from impact, *Comput Methods Appl Mech Eng* 394 (2022) 114913, doi:[10.1016/j.cma.2022.114913](https://doi.org/10.1016/j.cma.2022.114913).

- [67] T. Wu, A. Alshareef, J.S. Giudice, M.B. Panzer, Explicit modeling of white matter axonal fiber tracts in a finite element brain model, *Ann Biomed Eng* 47 (9) (2019) 1908–1922, doi:[10.1007/s10439-019-02239-8](https://doi.org/10.1007/s10439-019-02239-8).
- [68] T. Wu, F. Sato, J. Antona-Makoshi, L.F. Gabler, J.S. Giudice, A. Alshareef, M. Yaguchi, M. Masuda, S.S. Margulies, M.B. Panzer, Integrating human and nonhuman primate data to estimate human tolerances for traumatic brain injury, *J Biomech Eng* 144 (7) (2022) 071003, doi:[10.1115/1.4053209](https://doi.org/10.1115/1.4053209).
- [69] B. Yang, K.-M. Tse, N. Chen, L.-B. Tan, Q.-Q. Zheng, H.-M. Yang, M. Hu, G. Pan, H.-P. Lee, Development of a finite element head model for the study of impact head injury, *BioMed Res Int* 2014 (2014) e408278, doi:[10.1155/2014/408278](https://doi.org/10.1155/2014/408278).
- [70] X. Zhan, Y. Li, Y. Liu, A.G. Domel, H.V. Alizadeh, S.J. Raymond, J. Ruan, S. Barbat, S. Tiernan, O. Gevaert, et al., The relationship between brain injury criteria and brain strain across different types of head impacts can be different, *J R Soc Interface* 18 (179) (2021) 20210260, doi:[10.1098/rsif.2021.0260](https://doi.org/10.1098/rsif.2021.0260).
- [71] X. Zhan, Y. Liu, S. Raymond, H. Vahid Alizadeh, A. Domel, O. Gevaert, M. Zeineh, G. Grant, D. Camarillo, Rapid estimation of entire brain strain using deep learning models, *IEEE Trans Biomed Eng* 68 (11) (2021) 3424–3434, doi:[10.1109/TBME.2021.3073380](https://doi.org/10.1109/TBME.2021.3073380).
- [72] K. Zhang, L. Cao, A. Fanta, M.P. Reed, M. Neal, J.-T. Wang, C.-H. Lin, J. Hu, An automated method to morph finite element whole-body human models with a wide range of stature and body shape for both men and women, *J Biomech* 60 (2017) 253–260, doi:[10.1016/j.jbiomech.2017.06.015](https://doi.org/10.1016/j.jbiomech.2017.06.015).
- [73] W. Zhao, S. Ji, Mesh Convergence Behavior and the Effect of Element Integration of a Human Head Injury Model, *Ann Biomed Eng* 47 (2) (2019) 475–486, doi:[10.1007/s10439-018-02159-z](https://doi.org/10.1007/s10439-018-02159-z).
- [74] W. Zhao, S. Ji, White matter anisotropy for impact simulation and response sampling in traumatic brain injury, *J Neurotrauma* 36 (2) (2019) 250–263, doi:[10.1089/neu.2018.5634](https://doi.org/10.1089/neu.2018.5634).
- [75] W. Zhao, S. Ji, Displacement- and Strain-Based Discrimination of Head Injury Models across a Wide Range of Blunt Conditions, *Ann Biomed Eng* 48 (6) (2020) 1661–1677, doi:[10.1007/s10439-020-02496-y](https://doi.org/10.1007/s10439-020-02496-y).
- [76] Z. Zhou, X. Li, S. Kleiven, W.N. Hardy, Brain strain from motion of sparse markers, *Stapp Car Crash J* 63 (2019) 1–27, doi:[10.4271/2019-22-0001](https://doi.org/10.4271/2019-22-0001).
- [77] Zhou Z., Li X., Liu Y., Hardy W., Kleiven S. 2022. Brain strain rate response: addressing computational ambiguity and experimental data for model validation. [place unknown]. <https://doi.org/10.1101/2022.02.04.478773>