Accelerate Ped-Pro assessments using SimAI

Srikanth Adya, Akshay Jawale, Pierre Yser

Ansys

1 Abstract

Deep learning methods have had a significant impact on design process in the recent past. SimAI is a deep learning-based AI platform that has shown to be very effective in approximating the behavior of fluid flow applications, especially fully developed steady state flows simulated by CFD solvers. The underlying neural networks in SimAI are very versatile and can be easily extended to structural applications as well. This study aims at demonstrating the applicability of SimAI for non-linear transient structural simulations like pedestrian protection. We start with a simple Tube Crush model to demonstrate the use of SimAI to predict the deformed shape of the Tube at any time instance. We then train a model on different Tube shapes to show SimAI's ability to learn from non-parametric geometry. Finaly, we demonstrate how SimAI can be used to accelerate Ped-pro evaluations. The NCAC Accord model is used to generate 96 training points. This dataset is used to train a SimAI model and the resulting trained model can predict the full field hood deformation as well as the HIC value for the corresponding hit location within 10% relative error on any point on the vehicle hood. SimAI is many orders of magnitude faster in predicting the HIC than direct numerical simulation and hence can be very effective in evaluating designs upfront in the vehicle development process

2 Introduction

With shrinking automotive development timelines, CAE has been a very integral part of vehicle design process in the past couple of decades. Vehicle safety is one of the most critical paths in this process. With new safety regulations being added in multiple regions every year, virtual analysis has become a reliable tool to access a vehicle's crash and safety performance virtually, without having to rely too much on costly physical testing. LS-DYNA is one such tool that has become an industry standard when it comes to building high fidelity crash models. Over the years, the complexity of these models have grown exponentially with details like, airbags, dummies, restraint systems, spotwelds, adhesives, material failures etc. being captured to improve the predictivity of the model. While this has resulted in improving our confidence in CAE predictions and in turn reduction in the number of physical tests, it has also made the models much larger and thus increasing the need for large compute resources to solve these models. Off late, it is not uncommon to see full vehicle models with 40 to 50 million elements requiring about 25 to 30 hours to solve a complete crash event on about 700 to 1000 CPUs on a modern-day high-performance compute cluster.

Apart from the compute overhead, there is also a significant preprocessing cost involved in generating these models. Typically, once a new car design is available, it takes anywhere between two to three months to clean the CAD, mesh it, add all the necessary connections and materials, debug for any errors and establish a baseline performance. Usually, during the early concept phase of the design, the geometry is not fully mature. Parts and subsystems are borrowed from legacy programs and plugged into the new design. Since these components were not designed to be in the new vehicle, most often than not, a lot of time is spent in clearing geometric intersections and penetrations between neighbouring parts. These penetrations, if not cleared, would result in poor model predictions and numerical instabilities in the model that would cost a lot of time and effort during the design evaluations. And even if good clean geometry is available, most often, the CAE model fails to keep up with all the design changes and tends to lag the design iterations. Consequently, only few designs are ever thoroughly evaluated.

Hence, there is a need for a solution where a designer, who is not an expert in Finite Element Method, can quickly evaluate multiple designs and make an informed decision on the best option. Traditionally, optimization tools have tried to fill this need by parameterizing CAE models and building meta-models and response surfaces that can be used to predict the behaviour of a new design. But these meta-models rely heavily on parameterized input. It is easy to parameterize variables like part gauges and material properties but parameterizing the geometry is a challenge. Changing geometries needs to be done in conjunctions with a CAD tool or a morphing tool and this is usually a time-consuming process. Also, the response surfaces that are generated are usually only capable of predicting a scalar response like, the peak acceleration at a point in the model, maximum intrusion at a point in the vehicle or an injury measure on the dummy. Some of the newer tools also try to predict a signal instead of just a scalar, but even then, the useful information that can be extracted from these meta-models are restricted to a very specific region of interest in the vehicle. If the design change has an influence anywhere else in the model, the meta-model would fail to highlight it.

Hence there is a need for a tool that can quickly but accurately predict the full 3D response of a new design. With advances in machine learning methods, it is now possible to learn from lots of existing data and make inferences on unknown designs. SimAI is an Ansys tool that takes advantage of this technology and builds AI models that are trained on simulation data. In this paper, we discuss the technology being used as well as its application in vehicle crash performance predictions.

3 SimAI Technology

SimAI is a novel Deep Learning method which bakes in its proprietary architecture, state-of-the-art concepts like: INR, GNN, CNN, Multi-Scale learning, Fix point learning etc. . SimAI does not solve partial differential equations but learns the full 3D information coming from traditional solver solutions like LS-DYNA. It can then quickly infer new solutions for efficient design iterations. SimAI thus combines a unique Neural Network architecture with physical priors (integral coefficients computed from local values, strong spatial correlation, local invariance extraction …) to deliver AI models for any type of physics [1][2][3].

SimAI is delivered on a SaaS cloud platform that enables anyone to create AI models, store them as a library of models and to run inferences on these models to get high fidelity predictions for specific workflows. This platform is accessible through a webapp and through a Software Development Kit (SDK) allowing for agile workflow and process automation.

4 Numerical Model Description

SimAI can generate a reduced order model of both parametric and non-parametric data. In this section, three separate studies are described. The first case, an AI model is generated for a Tube crush model. This model has geometric variations that will be described in Section 4.1.1.

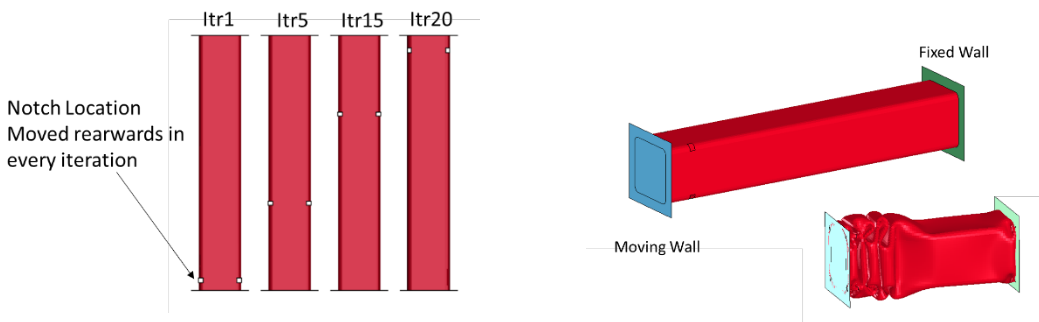


Fig 1. Tube with Notches

In Section 4.1.2, we extend this study to make the geometric variations a bit more pronounced by changing the cross-section of the tube. Finally in Section 4.2, we look at a pedestrian head impact model. In this case, the location of head-from is changed. Geometry of the vehicle remains the same but the boundary condition changes as the impact is evaluated at several locations on the hood as shown in Fig. 2. Hence boundary condition change is incorporated in the geometry. The AI model is generated to predict deformation on the hood as well as the corresponding HIC value.

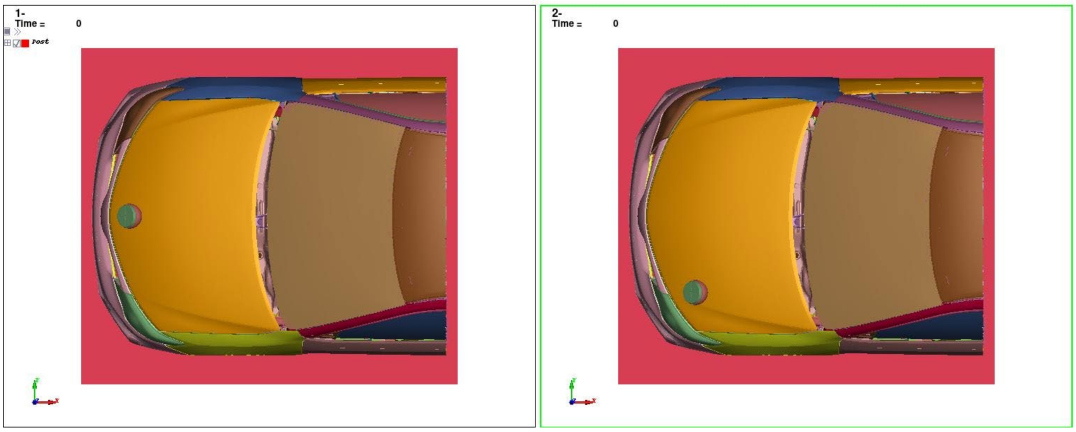


Fig.2 Difference in geometry due to change in position of Head

4.1 Tube Crush

4.1.1 Tubes with Notches

This dataset consists of a series of 20 rectangular tubes with notches as shown in Fig 1. Geometric variability is introduced in the model by moving the location of the notch rearwards in each design. One of the ends of the tube is constrained by an immovable rigid wall. The other end is in contact with a movable rigid wall which is prescribed an initial velocity of 10m/s. The material of the tube is assumed to be steel with a density of 7.8e-6, young's modulus of 210.0, yield strength 0.4 and tangent modulus of 2.0. The model uses kN, kg, ms and mm for unit system. The above image shows how the tube folds on itself when the rigid wall pushes against the tube.

4.1.2 Tubes with varying cross-section

In this dataset, the geometry variation is a little more pronounced by changing the cross-section of the tubes. Three different cross-sections, rectangular, circular and conical, were generated. Within each of these, tubes with three different dimensions were created as shown in the Table 1 and Fig 3. As in the first case, one end of the tube is resting against a fixed rigid-wall. The other end is in contact with a moving rigid-wall with an imposed initial velocity of 10m/s. The material properties are same as that used in the previous case.

Cross-section	**Set 1**	**Set 2**	**Set 3**
Conical	50mm front diameter	60mm front diameter	70mm front diameter
Circular	50mm	60mm	70mm
Rectangular	40mmx50mm	50mmx50mm	50mmx60mm

Table: 1. Tube Geometric Variations

Fig 3: Different geometries of the Tube Crush model.

4.2 Pedestrian Head Impact Model

Pedestrian protection assessment methods require multiple head impact assessments on a vehicle's hood and other front-end parts as shown in figure 4. The model shown below was derived from NCAC Accord model referenced here [4]. The Head injury (HIC value) is evaluated using the acceleration of the head form within a 15 or 36ms time window as shown in figure 4.

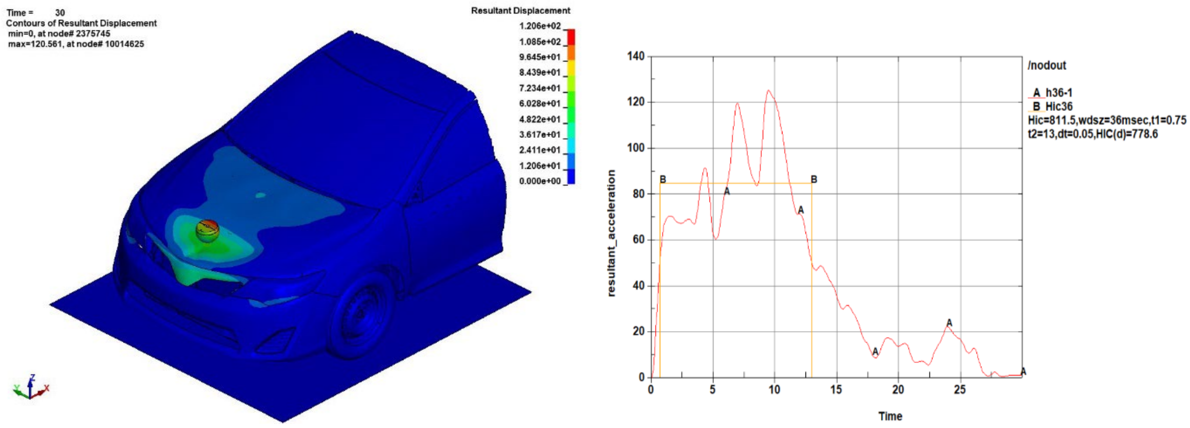


Fig.4 Pedestrian Head Impact deformation state and Acceleration-Time History plots

In this section, A child head form is impacted on Vehicle hood at an angle of 50 degree with horizontal at a speed of 40 KMPH. The HIC value is evaluated using Head acceleration using following formula,

$$HIC = \left\{(t_2 - t_1)\left[\frac{1}{t_2 - t_1}\int_{t_1}^{t_2} a(t)dt\right]^{2.5}\right\}_{max}$$

t_1, t_2 *are two instants which define start and end of recording, where value of HIC is maximum*

Total 106 simulation data points were generated by varying position of head across hood. Figure 4. shows typical hood deformation along with head acceleration-time history for one of the impact locations.

5 SimAI data Preparation and Model Training

In order to predict a transient response, we need to extract the output field of interest at different timesteps. The input data for SimAI needs to contain both the input geometry and the operating/boundary conditions. In our case, the boundary conditions are any thickness changes for the parts as well as the time at which the output response has been extracted. The boundary conditions are fed as a json file. For the tube crush cases, there is no part thickness variation. Hence the json file will only have time as a parameter. So is the case with the Pedestrian Impact model, json file contains time as a parameter. Input geometry which includes variation in head position is fed to SimAI. Fig 5. shows a typical boundary condition file.

```
{"Time": 29.99936866760254}
```

Fig 5: Boundary condition json file for pedestrian Head impact case consisting of time as parameter

Geometry is written out in .vtp or .vtu file formats. vtp holds the surface (shell elements) data and vtu holds the volume (solid elements) data in them. These files consist of the nodal coordinates, element connectivity along with any nodal or elemental outputs that can be used as inputs or outputs to the model. In the case of the tube crush models the .vtp file includes the undeformed shape and the displacement field at a given timestep. In case of Pedestrian Impact, the .vtp, .vtu files contain displacement field and the HIC field (HIC is single scalar value, hence it is mapped on the mesh as a constant field). The trained model is expected to predict back the displacement field and a HIC value for every hit location. To read the raw d3plot files and convert them to .vtp and .vtu format, pyDPF was used[5]. Fig 6. shows a sample vtp file for the tube crush and the pedestrian impact cases.

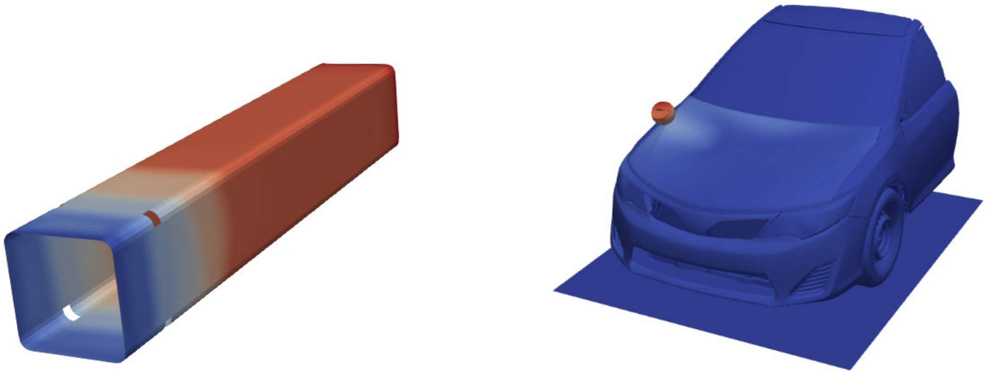


Fig 6: vtp files with displacement field on the nodes

For the Tube crush model with 20 different iterations and 22 states per iteration, extracting vtp for every state would result in a total dataset size of 440 data points. Similarly for pedestrian impact model, with 106 design points and 11 states per design point, we get a total of 1166 data points to train on. Once the input data is thus prepared, it is uploaded to SimAI platform. This can be done manually by dragging and dropping the dataset on the SimAI web interface or using a python SDK called pySimAI[6]. SimAI is currently available as a cloud-based solution and is hosted on AWS with all the necessary data security considerations. Once the data is uploaded the dataset within each project is then divided automatically by SimAI into two subsets as training set and test set.

- Training set is used to train the model
- Test set is used to assess the accuracy of the trained model. This unseen data gives an unbiased estimation of the model performance on new cases

5.1 Model configuration

The next step after the data is uploaded is Model Configuration. This involves identifying the inputs and the output for the AI model along with the boundary conditions. For the Tube Crush use cases, the inputs for the model are the geometry and the time step. Geometry is extracted from the uploaded .vtp file and the time is read from the boundary condition json file. UX, UY and UZ are the output displacement field that the model is trained to predict.

For the pedestrian head impact use case, time is read from .json file and geometry is extracted from .vtp file. Hence the geometry and time will serve as an input to SIMAI model. Also, along with the displacements UX, UY and UZ, HIC value is also defined as an output quantity. The model is now expected to learn the relationship between the geometry and the boundary conditions to the nodal displacements and the HIC value at any given time. Fig 7. shows the model configuration for the pedestrian impact model.

An integral global coefficient called HIC value is also defined for the pedestrian head impact model. This acts as a scalar quantity to compare between the solver and the predicted model and helps in accessing the accuracy of the model. This global coefficient acts as a metric and is not used to constraint the model during the training. The global coefficient is defined as

$$\frac{\int_s (HIC)dS}{\int_s (1)dS}$$

$$\frac{\int_s (HIC)dS}{\int_s (1)dS}$$

This just means that the constant HIC value that was defined on all the elements in the vtp file is summed up and is divided by the total surface area of the model, thus returning a scalar output.

The desired precision of the model is also selected in the model configuration. The precision level directly affect the time required to train the model. A model with accuracy set to “Precise” takes anywhwere upto 2 days to train. “Very Precise” takes upto a week to train.

6 Results and Discussion

All three models were trained on Tensor Core GPUs on SimAI with an accuracy level of “Precise”. In the following sections we will discuss the model accuracy as well as some observations on these models.

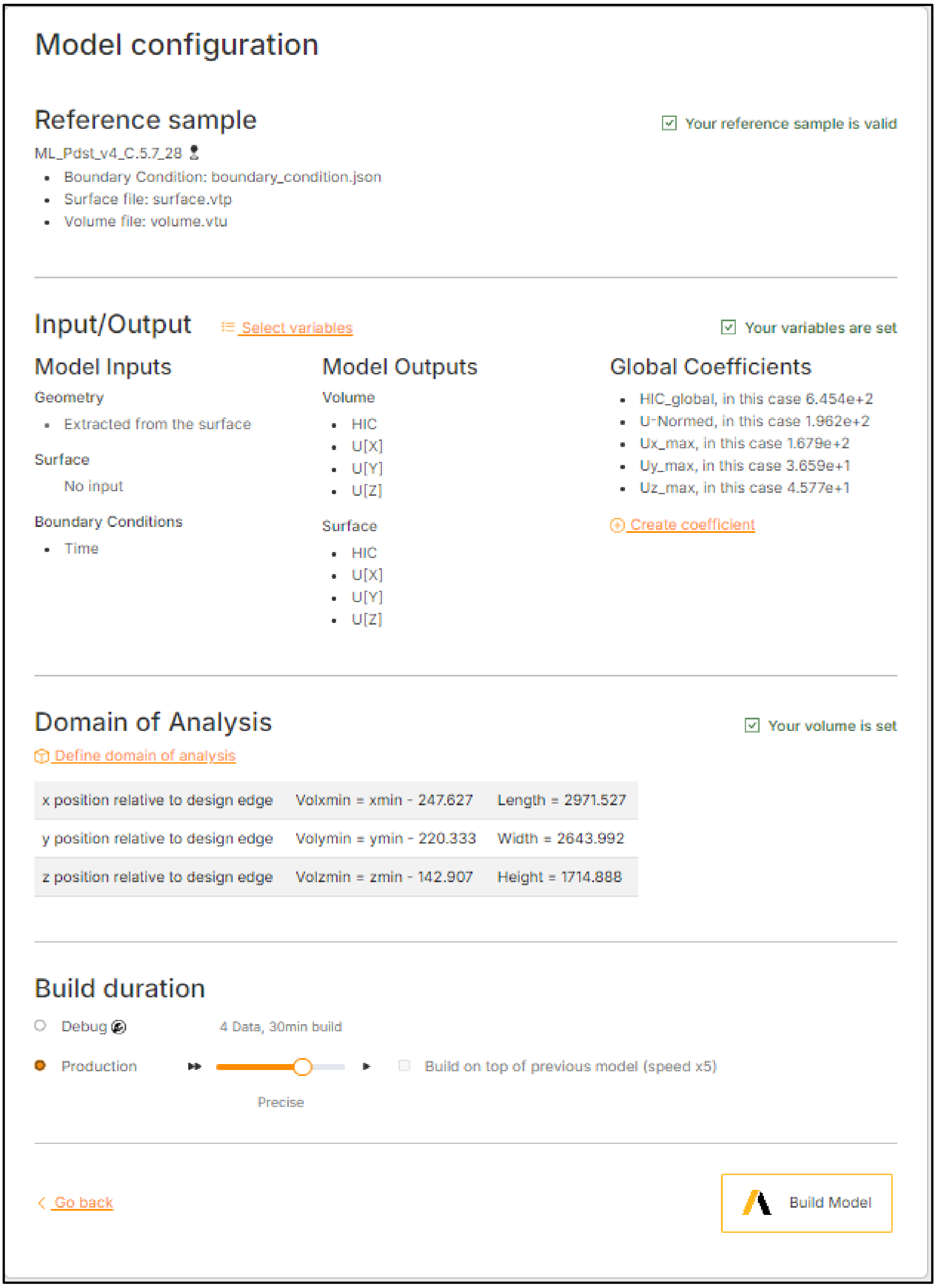
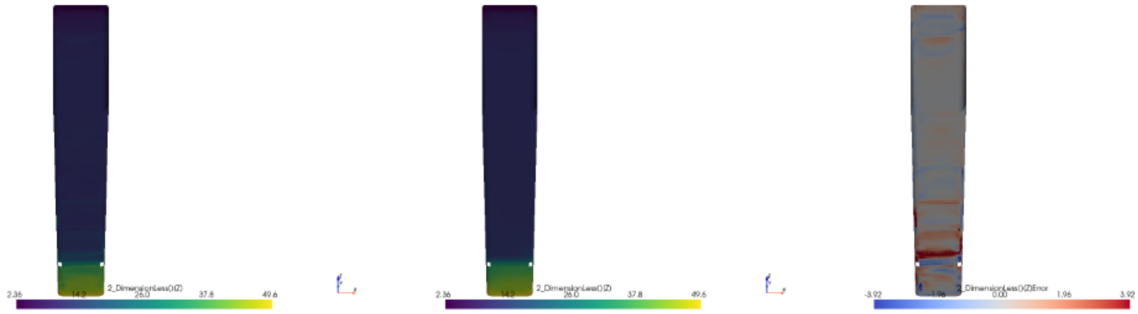


Fig 7: model configuration for Bumper Impact model

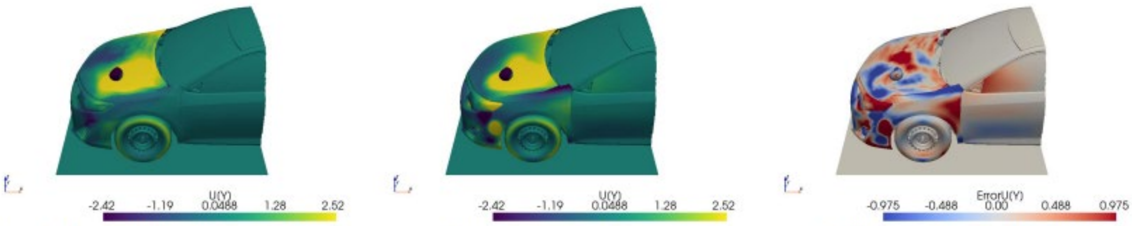
Test geometry 81



(a) SimAI prediction (b) Solver target (c) Difference

Fig 8 a) Tube Crush Case

3.4.3 Test geometry 63



(a) SimAI prediction (Solver scale) (b) Solver target (Solver scale) (c) Difference (Zoomed scale)

Fig 8 b) Pedestrian Head Impact Case

Fig 8: Surface field UZ predicted by SimAI, the solver target and the difference between them

6.1 Tube crush

6.1.1 Tube with Notches

This model took about 12hrs to train to the desired level of accuracy. Once the training is done, a model evaluation report is generated. The model evaluation report compares the SimAI prediction with the Solver simulation and highlights the regions where there are differences as shown in Fig 8. It can also be seen from the Fig 9. that SimAI has managed to learn the relationship between the location of the notch on the tube and the crush pattern of the tube. SimAI accurately predicts the buckling of the tube to be at the location where the notch exists. As can be seen, the AI model has learnt that the buckling of the crush tube occurs where the notch is and hence the tube buckles in the front for the top model corresponding to the notch being in the front. Similarly, the tube buckles at the rear end in the bottom model corresponding to the notch being at the rear end in this case. Model also predicts well for any other location of the notch along the length of the tube.

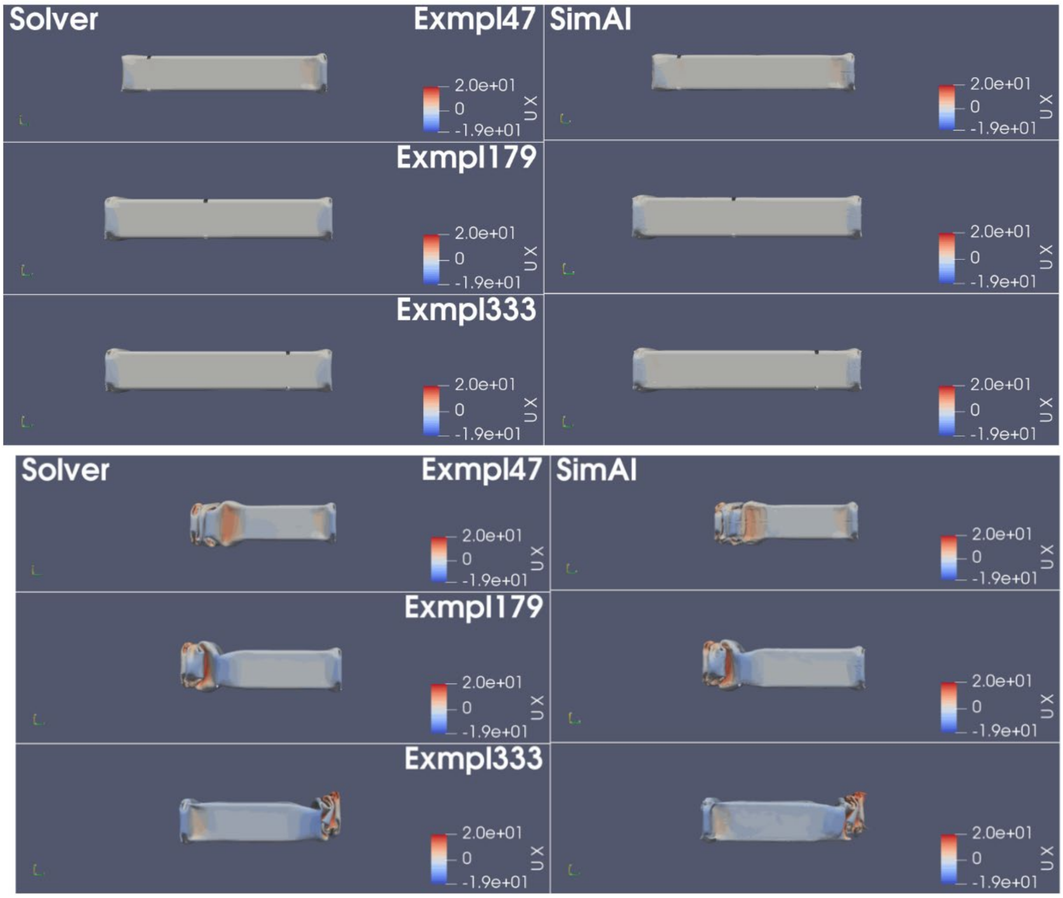


Fig 9: Crush behaviour at different timesteps comparison between SimAI and Solver

6.1.2 Model prediction limit

Now that the model behaves well on different geometries within the design space, one more evaluation was done to see how well the model can generalize on a significantly different geometry. In this case, the model was asked to predict the crush on a Tube with a considerably long notch compared to all other samples in the training data. SimAI model predicted a similar crush pattern to the previous cases whereas the solver output is significantly different. Fig 10. shows the comparison of SimAI prediction to solver output for this long notch case. The buckling of the model at the mid-section due to the long notch is not predicted by SimAI. This is because SimAI does not solve for any physics equations but infers the full 3D information from the training dataset. Since no such example existed in the training dataset, SimAI could not predict the true behaviour for the model.

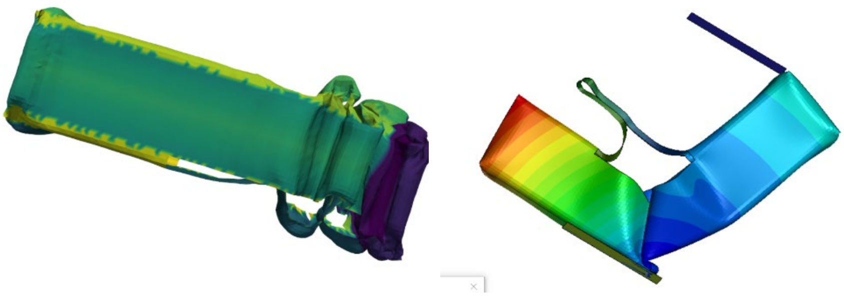


Fig 10: Crush behavior SimAI vs Solver long notch

6.1.3 Retraining the model with the new design

The model was then retrained by adding the long-notch example into the training dataset. As before, the d3plots for this new case had 20 states. Each state along with the corresponding displacement field was extracted as a vtp file and uploaded to SimAI. The retrained model is now capable of predicting the behaviour of the long notch more accurately. The model predicts the base case of small-notch as well as generalizes to the long-notch as can be seen in Fig 11.

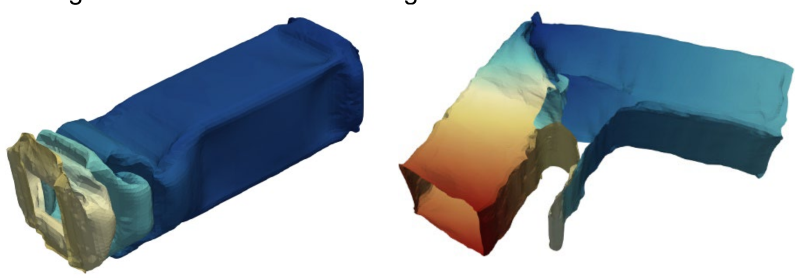


Fig 11: Retrained model with the long-notch sample added to training set. Model now predicts both cases reasonable accurately

6.1.4 Tubes with varying cross-section

The AI model in this case was trained on different cross-sectional tubes along with changing dimensions. For this case, the model was able to generalize well based on the input geometry. The crush pattern predicted by SimAI for a given shape and dimension agrees well with the solver predictions as shown in Fig 12.

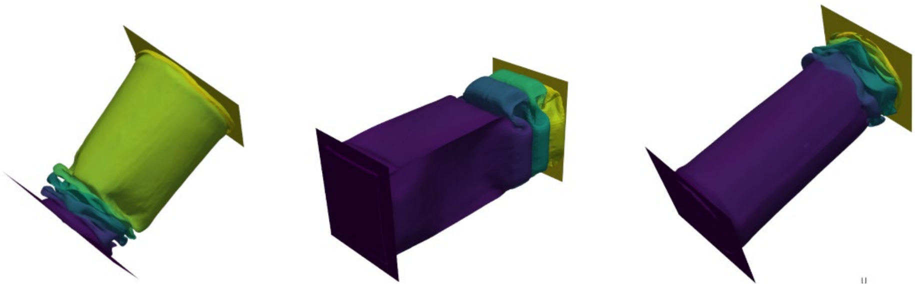


Fig 12: SimAI predictions for differently shaped crush tubes

6.2 The Pedestrian Head Impact Case

The aim of this load case was to predict both the deformed shape as well as the HIC value for a given geometry. This model took about 20hrs to train on NVIDIA A10G Tensor Core GPU. Fig 12. shows the trend plot for the global coefficient "HIC_global". The X-axis is the Solver prediction for a design at a given timestep. Y-axis is the corresponding SimAI prediction for the same design at the same time step. A perfect AI model would have all the points on the diagonal meaning a 100% accuracy. But that would also indicate that the model is overfitting to the training data and might not generalize well. As seen in the Fig 13. SimAI model has most of the predictions within 10% error band indicating that model is generalizing well. The model evaluation report also generates a difference plot of the displacement field prediction as shown in Fig 8b. Fig 14 highlights the comparison of the displacement field between SIMAI prediction Vs Solver (ground truth in this case)

The SimAI model can be used to make multiple predictions on a new geometry at different timesteps. Each of these predictions will predict a displacement field which can be overlayed on the undeformed geometry to produce the deformed shape. These different states can be then stitched together to generate the time varying displacement field for the entire crash event.

Finally, each prediction also generates the global coefficient "HIC_global". Fig 15. compares the HIC value predictions between Solver and SimAI. Again, the AI model predictions compare reasonably well with the Solver.

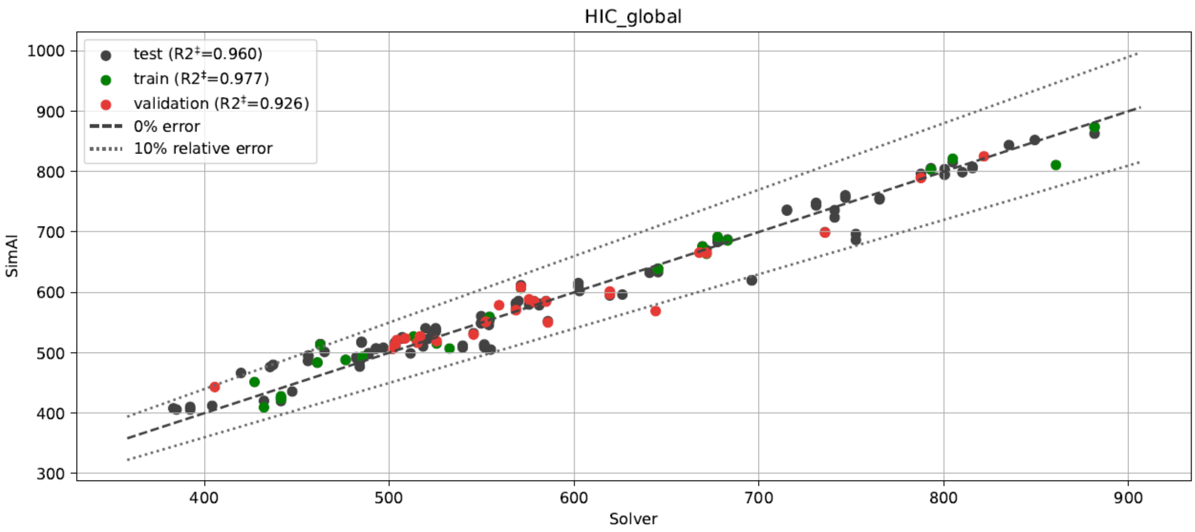


Fig 13: SimAI trend plot for HIC value

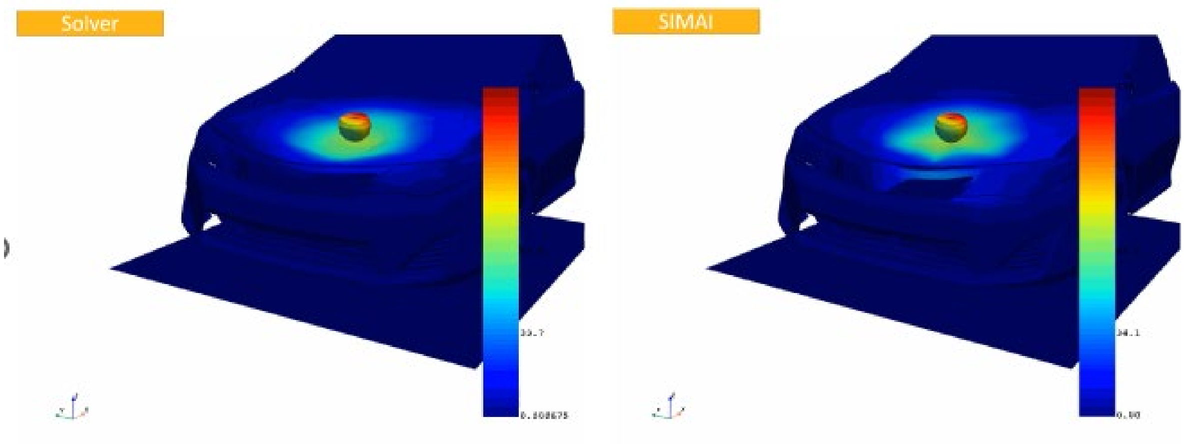


Fig 14: Resultant Displacement prediction comparison between SIMAI and Solver

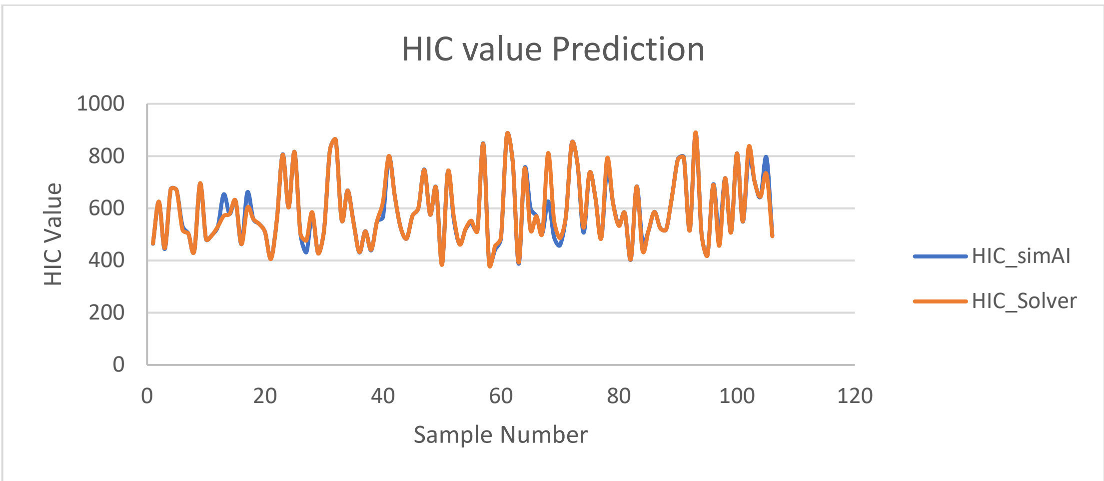


Fig 15: Comparison of HIC value prediction between Solver and SimAI

The predictions made on the AI models are orders of magnitude faster than the Solver. Each of these predictions take less than a second for HIC value prediction and about 10s to predict the entire displacement field and post-processing it. Predictions can also be made at any time within the lower and upper bounds of the time within the dataset. Thus, the SimAI predictions can be of much finer sampling compared to the Solver.

7 Conclusions

The above set of use cases demonstrate the applicability of Deep Learning technology for generating AI models for structural applications by learning the physical behaviour from past LSDYNA simulations. The models can predict both the final deformed state as well as any other intermediate states, thus making it possible to have one single AI model for the transient non-linear event.

The ability of the model to learn the geometric variations between different designs without having to parameterize the geometry makes it very useful for quickly analysing new designs. The study highlights the effectiveness of the AI models as well as its limitations with regards to the model generalization, on completely different geometry than the ones the model was trained on.

The AI-simulation process is very simple and fast to deliver full field responses on surfaces and volumes. Once the model is trained, non-CAE engineers can use the trained model to make predictions on new designs. The predictions can be made on CAD (stl files) directly, thus avoiding the need for meshing the CAD.

8 References

[1] Dr. J. Verriere et. al, “How to accelerate car aerodynamics thank to deep learning physics”
[2] Dr. J. Verriere et. al, “Deep Learning Physics for hydrodynamic of trading vessels"
[3] Louis Serrano et al, “Neural Field Modeling for Reynolds-Averaged Navier-Stokes Equations”
[4] https://www.nhtsa.gov/document/honda-accord-lsdyna-zip
[5] https://dpf.docs.pyansys.com/version/0.12/getting_started/index.html
[6] https://simai.docs.pyansys.com/version/stable/
[7] LS-DYNA Keyword User Manual – https://lsdyna.ansys.com/manuals/