Reproducibility Report for Zeng, Rezaei, Carrillo, Davidson, Xu, Banerjee, and Ding, 2024, "Chemomechanical damage prediction from phase-field simulation video sequences using a deep-learning-based methodology", *iScience*, Volume 27, Issue 9, pp.110822.

Prepared by Quan Zeng

January 16, 2025

1 Computer and software environment

All code for this study was developed and executed using Python 3.8.2. To ensure consistency and reproducibility, specific versions of Python libraries were utilized throughout the analysis. Key libraries included OpenCV (version 4.6.0.66) for image processing and TensorFlow (version 2.9.1) for model training. Data handling and manipulation were performed using NumPy (version 1.20.1) and Pandas (version 1.2.4), while Matplotlib (version 3.3.4) was employed for generating visualizations.

The computational analysis was conducted on the Grace cluster at Texas A&M University's High Performance Research Computing (HPRC) facility. The resources allocated for this analysis included one GPU node with access to two A100 GPUs (each with 40 GB of memory) and 112 GB of system memory, with each node having a total of 384 GB of memory.

2 Explanations of the data files

The folder **data.video** contains the video files obtained from the phase-field simulation. As outlined in the subsection"Results – Damage prediction using DL models – Data description and preprocessing", there are eight damage videos and eight stress saved as AVI files (damage[0..90]degrees.avi, hydrostatic[0..90]degrees.avi), corresponding to eight grain orientations 0°, 10°, 25°, 30°, 45°, 60°, 75°, and 90°, respectively.

For further analysis, each video was converted into NPY format (damage[0..90]degrees.npy, hydrostatic[0..90]degrees.npy) and stored in the folder **data.npy**. The code to reproduce this conversion is in "ConvertAVItoNPY.py".

Table 1 summaries the data files and their names in each data folder.

Table 1: Data files and names

Data folder name	**Number of files**	**File names**
data.video	16 avi files	damage[0..90]degrees.avi, hydrostatic[0..90]degrees.avi
data.npy	16 npy files	damage[0..90]degrees.npy, hydrostatic[0..90]degrees.npy

3 Reproducing the results in the main paper

Table 2: Reproducing the results in the paper

To reproduce	**Input data file**	**Code file**	**Output**
Figure 4	data.video/damage30degrees.avi, data.npy/damage30degrees.npy	CropSnapshot.py	Two PNG figure files of Figure 4 Middle, and Right
Figure 5A	data.npy/damage[0..90]degrees.npy, data.npy/hydrostatic[0..90]degrees.npy	DmgBinTrain0.py, DmgBinTest0.py, EnsBinTrain0.py, EnsBinTest0.py, utils.py	Figure 5A damage model output in a PNG file, Figure 5A ensemble model output in a PNG file

Figure 5B	data.npy/damage[0..90]degrees.npy, data.npy/hydrostatic[0..90]degrees.npy	DmgBinTrain60.py, DmgBinTest60.py, EnsBinTrain60.py, EnsBinTest60.py, utils.py	Figure 5B damage model output in a PNG file, Figure 5B ensemble model output in a PNG file
Figure 6A	data.npy/damage[0..90]degrees.npy, data.npy/hydrostatic[0..90]degrees.npy	DmgOrgTrain0.py, DmgOrgTest0.py, EnsOrgTrain0.py, EnsOrgTest0.py, utils.py	Figure 6A damage model output in a PNG file, Figure 6A ensemble model output in a PNG file
Figure 6B	data.npy/damage[0..90]degrees.npy, data.npy/hydrostatic[0..90]degrees.npy	DmgOrgTrain60.py, DmgOrgTest60.py, EnsOrgTrain60.py, EnsOrgTest60.py, utils.py	Figure 6B damage model output in a PNG file, Figure 6B ensemble model output in a PNG file
Figure 7	data.npy/damage60degrees.npy, data.npy/hydrostatic60degrees.npy	CtrPlot.py, utils.py	Figure 7 in six PNG files
Table 1 0°	data.npy/damage0degrees.npy, data.npy/hydrostatic0degrees.npy	CtrPlot.py, utils.py	A TXT file for Table 1 0°
Table 1 60°	data.npy/damage60degrees.npy, data.npy/hydrostatic60degrees.npy	CtrPlot.py, utils.py	A TXT file for Table 1 60°