Data-Driven Nonlinear Deformation Design of 3D-Printable Shells - Supplementary Material

Samuel Silverman[1] Kelsey L. Snapp[2] Keith A. Brown[2,3,4] Emily Whiting[1]

1 Performance Data Preprocessing

We employed a data preprocessing pipeline to turn generated cylindrical shell (GCS) force-displacement curves into condensed performance vectors. We addressed instrument noise inherent in the raw force-displacement data from compression testing. The data contains hundreds of force measurements per millimeter at irregular displacement intervals. We used a median filter followed by a mean filter, each with a window size of 20 measurements. To ensure size consistency, we converted all force-displacement curves to 100 points at evenly-spaced displacement values. For each point, we selected the nearest force sample after filtering.

Before applying principal component analysis (PCA), we log-scaled and standardized the force values across all curves. To determine the efficacy of PCA, we looked at a scree plot (Figure S1) of the percentage of the cumulative variance in the data. Using the elbow method, we selected ten principal components accounting for approximately 99.8% of the cumulative explained variance. Figure S2 provides a visualization of a curve and the decomposed principal components.

Our performance vectors, denoted as $\mathbf{p} \in \mathbb{R}^{11}$, combine these ten principal component coefficients with the maximum displacement value. We standardized the maximum displacement values across all curves.

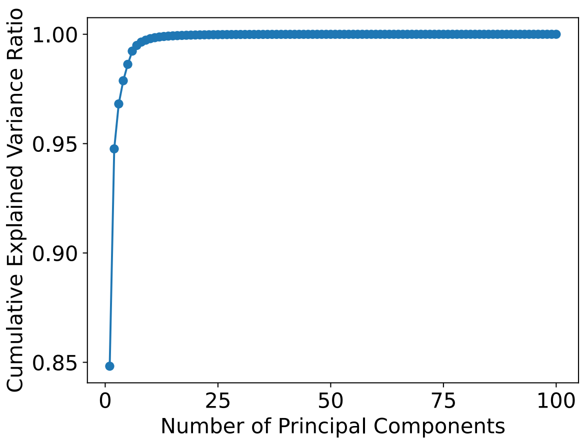


Figure S1: **PCA scree plot.** Cumulative explained variance for the 100 force principal components.

2 GCS Finite Element Analysis

We performed a simulated compression test of a GCS part using the finite element method implemented in Abaqus. We discretize the GCS into 7631 quadratic tetrahedral elements and model the compression plates as non-deformable cylinders.

[1]Department of Computer Science, Boston University, Boston, MA, USA.
[2]Department of Mechanical Engineering, Boston University, Boston, MA, USA.
[3]Division of Materials Science & Engineering, Boston University, Boston, MA, USA.
[4]Physics Department, Boston University, Boston, MA, USA.

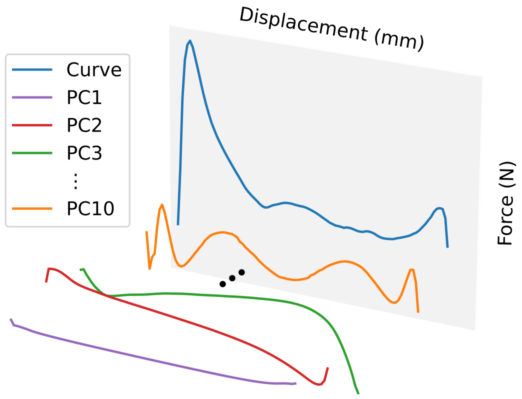


Figure S2: **Force decomposition.** Principal Component Analysis (PCA) decomposes the force values for a force-displacement curve (blue) into principal component force curves. Prior to PCA, the force values are log-scaled and standardized.

We create an elastoplastic material mimicking PLA using a stress-strain curve from a compression test performed on a cylinder PLA sample. The material's elastic behavior is defined by Young's modulus of 1390 MPa and Poison's ratio of 0.36 [1]. Its plastic behavior is defined by seven yield stress-plastic strain points. Additionally, we have set the coefficient of friction between the compression plate and PLA to 0.4 [1].

3 Application Details

3.1 Impact Absorption

For the egg drop test, we dropped United States Department of Agriculture large eggs onto a pad that must absorb the impact energy generated during the drop without exerting a force significant enough to cause the eggs to break. We used a gripper controlled by an Arduino Uno to drop eggs from a consistent height of 50 cm.

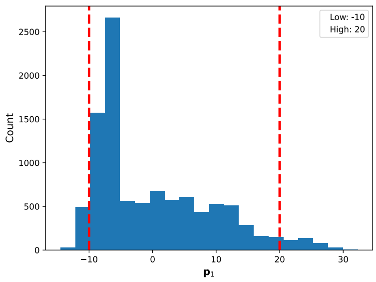
We defined two metrics for impact absorption: the target energy absorption E and a force threshold F. We estimated the mass per egg to be 60 g and calculated the impact absorption E as

$$E = mgh = 0.06 \text{ kg} \cdot 9.807 \text{ m/s}^2 \cdot 0.5 \text{ m} = 0.294 \text{ J},$$

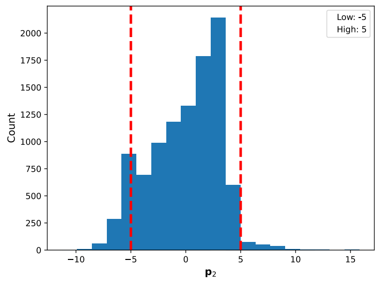
and defined $F = 40$ N[2]. Our pad design required four GCS arranged in a 2×2 array. Therefore, E and F are divided by four to get the appropriate values per GCS ($E = 0.0735$ J and $F = 10$ N).

We optimized for a valid performance vector $\mathbf{p}$ that best meets the user-specified metrics to use as input to inverse design. Note that $\mathbf{p}$ contains the ten force principal component coefficients with the standardized last displacement value. We determined the validity of $\mathbf{p}$ using the values found in the dataset (Figure S3). We used the dual annealing algorithm from SciPy [3] using the default parameters. In practice, a maximum number of global search iterations (`maxiter` parameter) of 250 provided an acceptable tradeoff of speed and accuracy. We measured the average runtime of dual annealing for the impact absorption optimization to be 1.9 ± 0.23 s for 20 initializations.

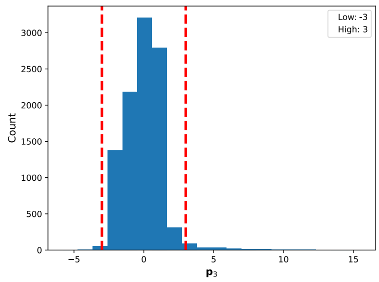
To evaluate the optimized pad's performance, we compared its energy-absorbing capabilities to (1) a pad with unoptimized GCS and (2) no pad. Notably, the optimized pad demonstrated a 100% survival rate, while the unoptimized pad exhibited a significantly lower 20% survival rate, followed by a 0% survival rate with no pad. Figure S4 provides visual examples of egg impacts on both pad types, illustrating the contrast in their protective capabilities.



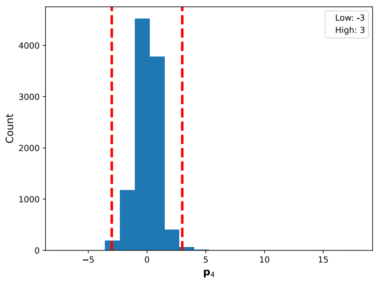
(a) $\mathbf{p}_1$ **distribution.**



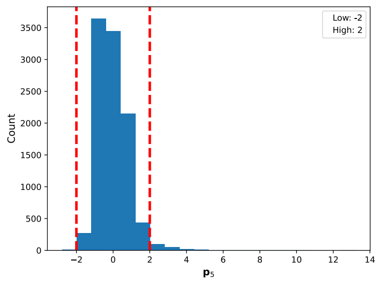
(b) $\mathbf{p}_2$ **distribution.**



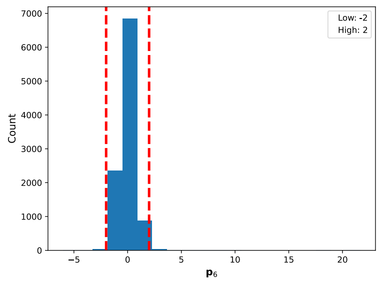
(c) $\mathbf{p}_3$ **distribution.**



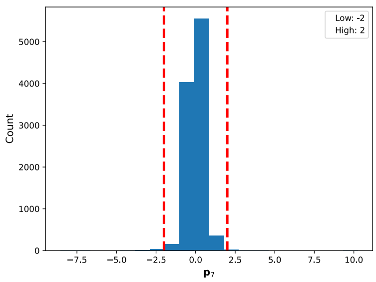
(d) $\mathbf{p}_4$ **distribution.**



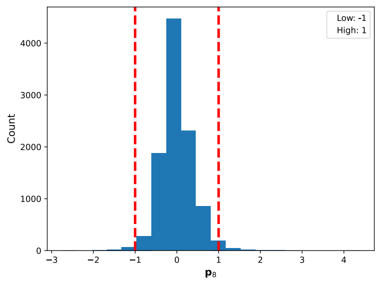
(e) $\mathbf{p}_5$ **distribution.**



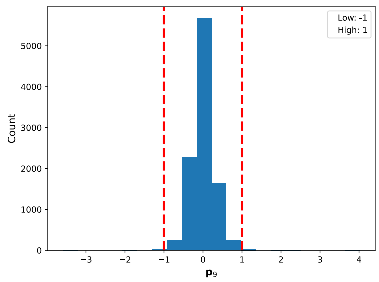
(f) $\mathbf{p}_6$ **distribution.**



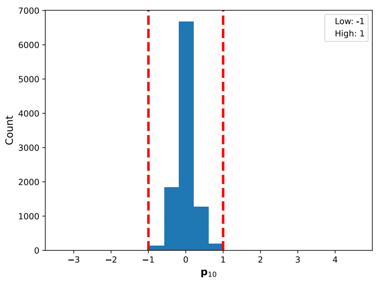
(g) $\mathbf{p}_7$ **distribution.**



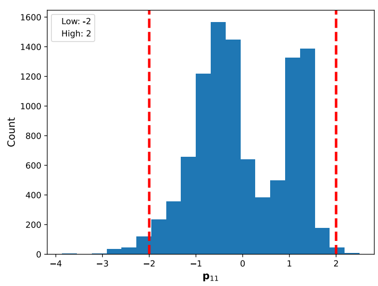
(h) $\mathbf{p}_8$ **distribution.**



(i) $\mathbf{p}_9$ **distribution.**



(j) $\mathbf{p}_{10}$ **distribution.**



(k) $\mathbf{p}_{11}$ **distribution.**

Figure S3: **Performance vector validity.** The distribution of performance vector $\mathbf{p}$ values found in the dataset. Red lines indicate the range we selected as valid used in the impact absorption force-displacement curve optimization. Note, $\mathbf{p}_1, \ldots, \mathbf{p}_{10}$ are the force principal component coefficients and $\mathbf{p}_{11}$ is the standardized last displacement value.

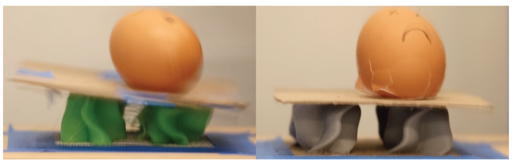


Figure S4: **Impact-absorbing pads.** The pad with GCS optimized for the egg drop test (left) absorbed the impact energy without breaking the egg. On the contrary, the pad with GCS not optimized for the egg drop test (right) exerted a force large enough to break the eggshell.

3.2 Material Emulation

We created GCS designs that replicate the behavior of polyurethane (PUR) foam, a material commonly employed in packaging. We used 32 custom parts to line an 8 in $\times$ 8 in $\times$ 3 in cardboard box: 16 for the base and four per side.

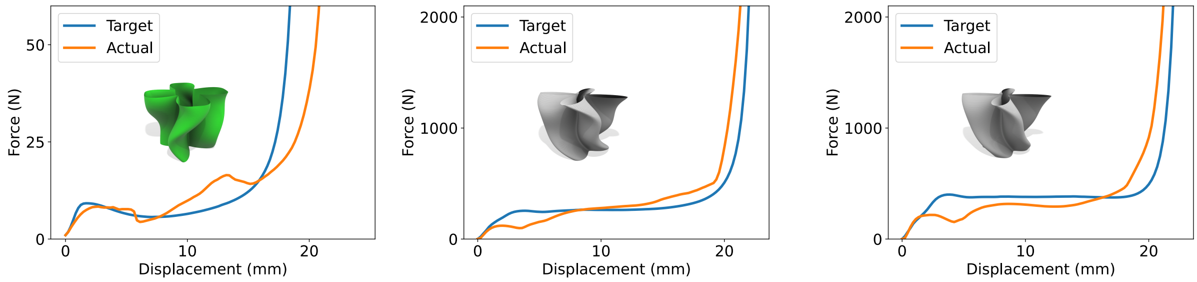
The parts for the base and sides emulate different volumes of PUR foam. Therefore, separate parts are needed. We arrived at target force-displacement curves for each part through conversion from a stress-strain curve of PUR foam,

$$\text{force} = \sigma \times \text{area}, \qquad \text{displacement} = \varepsilon \times \text{height}, \tag{1}$$

where σ and ε are the PUR foam's stress and strain, respectively.

3.3 Physical Evaluation

We obtained the actual force-displacement behavior for each application's generated GCS design through fabrication and testing (Figure S5).



(a) **Egg drop test.** (b) **Packaging (base).** (c) **Packaging (side).**

Figure S5: **Applications physical evaluation.** We fabricated and tested each application's generated GCS design to obtain the experimental force-displacement curves (orange). For reference, we provide the target force-displacement curves (blue).

References

[1] Pastor-Artigues MM, Roure-Fernández F, Ayneto-Gubert X, Bonada-Bo J, Pérez-Guindal E, Buj-Corral I. Elastic Asymmetry of PLA Material in FDM-Printed Parts: Considerations Concerning Experimental Characterisation for Use in Numerical Simulations. Materials. 2020;13(1). Available from: `https://www.mdpi.com/1996-1944/13/1/15`.

[2] Jianguo Z, Fang W, Shugang Y, YiNong Z, Haiyan S, Maoting L. Analysis of Drop Impact Behaviour of Eggs with Different Drop Height and Cushion Materials. In: Proceedings of the AASRI International Conference on Industrial Electronics and Applications (2015). Atlantis Press; 2015. p. 36-9. Available from: `https://doi.org/10.2991/iea-15.2015.10`.

[3] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods. 2020;17:261-72.