


Prediction and control of fracture paths in disordered architected materials using graph neural networks

Journal Article**Author(s):**

Karapiperis, Konstantinos; Kochmann, Dennis M. 

Publication date:

2023-06-02

Permanent link:

<https://doi.org/10.3929/ethz-b-000615102>

Rights / license:

[Creative Commons Attribution 4.0 International](#)

Originally published in:

Communications Engineering 2(1), <https://doi.org/10.1038/s44172-023-00085-0>

Funding acknowledgement:


101024077 - Data-Driven Design of Disordered Materials (EC)



<https://doi.org/10.1038/s44172-023-00085-0>

OPEN

Prediction and control of fracture paths in disordered architected materials using graph neural networks

Konstantinos Karapiperis¹ & Dennis M. Kochmann ¹✉

Architected materials typically rely on regular periodic patterns to achieve improved mechanical properties such as stiffness or fracture toughness. Here we introduce a class of irregular cellular materials with engineered topological and geometrical disorder, which represents a shift from conventional designs. We first develop a graph learning model for predicting the fracture path in these architected materials. The model employs a graph convolution for spatial message passing and a gated recurrent unit architecture for temporal dependence. Once trained on data gleaned from experimentally validated elastoplastic beam finite element analyses, the learned model produces accurate predictions overcoming the need for expensive finite element calculations. We finally leverage the trained model in combination with a downstream optimization scheme to generate optimal architectures that maximize the crack path length and, hence, the associated fracture energy.

¹Mechanics & Materials Lab, Department of Mechanical and Process Engineering, ETH Zurich, 8092 Zurich, Switzerland. ✉email: dmk@ethz.ch

Predicting the propagation of cracks in cellular and lattice materials is generally considered a challenging task, especially in the presence of disorder¹. Yet, understanding the process of crack growth is important in many contexts, such as in creating new tougher materials or in controlling the propagation of cracks away from potentially critical structural elements². In fact, with the manufacturing-enabled rise of architected materials made of metals, polymers, and ceramics^{3–5}, the study of their fracture properties has gained additional attention.

Despite significant theoretical progress, recent studies have outlined fundamental challenges. Concepts traditionally borrowed from linear elastic fracture mechanics (LEFM), such as the stress intensity factor and corresponding fracture toughness⁶, have been found insufficient to characterize fracture in architected materials^{7,8}. In the presence of ductility, characterizing the crack tip field⁹ presents additional challenges given the extent of the plastic/process zone and its dependence on the unit cell topology^{10,11}. Moreover, while most prior works have focused on periodic cellular materials, little do we know about the influence of aperiodicity, for example, through the influence of imperfections^{12,13}, irregularity^{1,14}, and finite size^{8,15}, on fracture behavior.

Physics-based models of fracture, although predictive, are often computationally expensive^{11,16–19}, which hinders their application in understanding the above effects, let alone exploiting them in material design and optimization²⁰. On the other hand, machine learning has proved successful in learning complex mechanical phenomena²¹, and it has recently also been applied to the problem of fracture, predominantly for the purpose of building predictive surrogate models. For example, recent studies^{22,23} predicted fracture paths in brittle materials. Yet, those were limited to predicting whether or not preexisting cracks coalesce—a classification task. The latter was augmented by a prediction of the propagating crack tip position between coalesced cracks in ref. ²⁴. Convolutional neural networks to predict crack paths were developed in refs. ^{25,26} and, more recently, have been combined with long short-term memory (LSTM) architectures in refs. ^{18,27,28}. Similarly, machine learning methods have been developed to learn fracture paths in atomistic systems^{29,30} as well as to study the influence of defects on the failure of graphene³¹ and silica glass³². Note that most of these approaches depend on convolutions, which are restricted to a Cartesian representation of the domain of interest, and they do not directly apply to cellular materials.

Going beyond fracture prediction, a few efforts have been reported that aimed at engineering the fracture path in materials by exploiting, e.g., hierarchy³³, multiple phases^{34,35}, and grain boundaries³⁶. In terms of design approaches, genetic algorithms have been developed for polycrystalline solids³⁷, while greedy methods have been utilized in the design of pixel-based materials³⁸. Most studies related to the design of periodic cellular materials have been restricted to heuristic or bioinspired approaches, e.g., by introducing a regular pattern of topological alterations², creating interpenetrating phases³⁵, or by engineering the strut thickness in honeycombs and creating softer struts through which the cracks can propagate³⁹. The use of optimization-based techniques to control the fracture path in cellular materials or the explicit incorporation of disorder as a design tool have both remained relatively unexplored.

In this work, we develop a graph neural network-based surrogate model to predict the propagation of cracks in cellular materials. The model works by learning a mapping between the local geometry/topology near the crack tip and the incremental crack advance. We demonstrate this methodology on Voronoi architectures ranging from near-perfectly ordered to highly disordered. A comprehensive dataset of these architectures is

generated by sampling from a carefully constructed statistical ensemble. We develop and validate—using simple tests of 3D-printed specimens—a reduced-order elastoplastic beam model, which serves as the basis for creating a large dataset of mode-I fracture simulations. The latter is used to train a machine learning model whose architecture combines graph convolutions and gated recurrent units. Once its prediction accuracy is established, the machine-learned model is leveraged in a downstream optimization task to maximize the length of the fracture path and the associated fracture energy.

Results

Disordered cellular solids. Toward a versatile albeit simple design of disorder cellular solids, we focus on Voronoi tessellations in 2D, i.e., on truss networks generated by partitioning space into cells, each surrounding a nucleus. Such tessellations are prevalent in nature (Fig. 1a). The cells' edges define sets of points that are equidistant to the two closest nuclei (Fig. 1b). To extend the classical construction of Voronoi networks (see, e.g., ref. ⁴⁰) into a more powerful and descriptive approach, we consider a statistical ensemble of those networks with a controllable range of geometrical and topological disorder. We define the topological disorder H_1 as the variance of the local coordination number Z (number of neighbors of a Voronoi cell), i.e., $H_1 = \langle Z^2 \rangle - \langle Z \rangle^2$. An interpretation is given in Fig. 1b, which shows cells colored according to their coordination number as well as their statistical distribution in the whole domain. Similarly, we define the geometrical disorder H_2 as the variance of the internuclei distance, i.e., $H_2 = \langle r_{ij}^2 \rangle - \langle r_{ij} \rangle^2$, where i, j denote neighboring nuclei (Fig. 1b). We associate $H_1, H_2 = 0$ with a regular hexagonal honeycomb. Note that this is a design choice; we could alternatively consider, e.g., a perfect square tessellation. Choosing increasing values of H_1, H_2 leads to increasingly disordered designs. As outlined in “Methods”, we proceed to generate a statistical ensemble, sampling from which results in a dataset of roughly 50,000 architectures, each comprised of $N = 162$ nuclei and ranging from perfectly ordered ($H_1, H_2 \rightarrow 0$) to highly disordered ($H_1, H_2 > 1$). The histogram of the corresponding topological and geometrical disorder in the dataset is shown in Fig. 1c, illustrating the number of realizations sampled for each combination. Note that due to the interplay of topological and geometrical disorder, one cannot populate the entire H_1 - H_2 space. Instead, for a given H_2 , a certain maximum value of H_1 can be accommodated. This notion is reflected in the inadmissible region shown in Fig. 1c.

Mechanical properties. We consider the canonical mode-I fracture problem^{7,29,41}, as shown in Fig. 2a, which involves a sample loaded vertically in tension. This is the simplest setting that abides by the ASTM standards⁴² and is used here to reveal the physics of the fracture process. A notch is placed at the left side of a rectangular sample (amounting to 25% of its width) by removing any intersecting struts. To minimize boundary effects, we keep one layer of cells around the exterior boundary at their regular positions (effectively reducing the disorder to zero in this thin zone). The upper and lower boundaries are displaced vertically at a constant rate, leading to the opening of the crack.

Simulations of such systems typically either rely on fully resolved finite element (FEM) calculations³⁹, which results in prohibitive computational costs for producing a sufficiently large dataset for learning, or simple strut failure criteria based on maximum pointwise stress or strain³⁵, which may result in an oversimplification of the mechanics. To mitigate these issues, we develop an accurate albeit efficient reduced-order corotational beam model that captures the essential underlying mechanics.

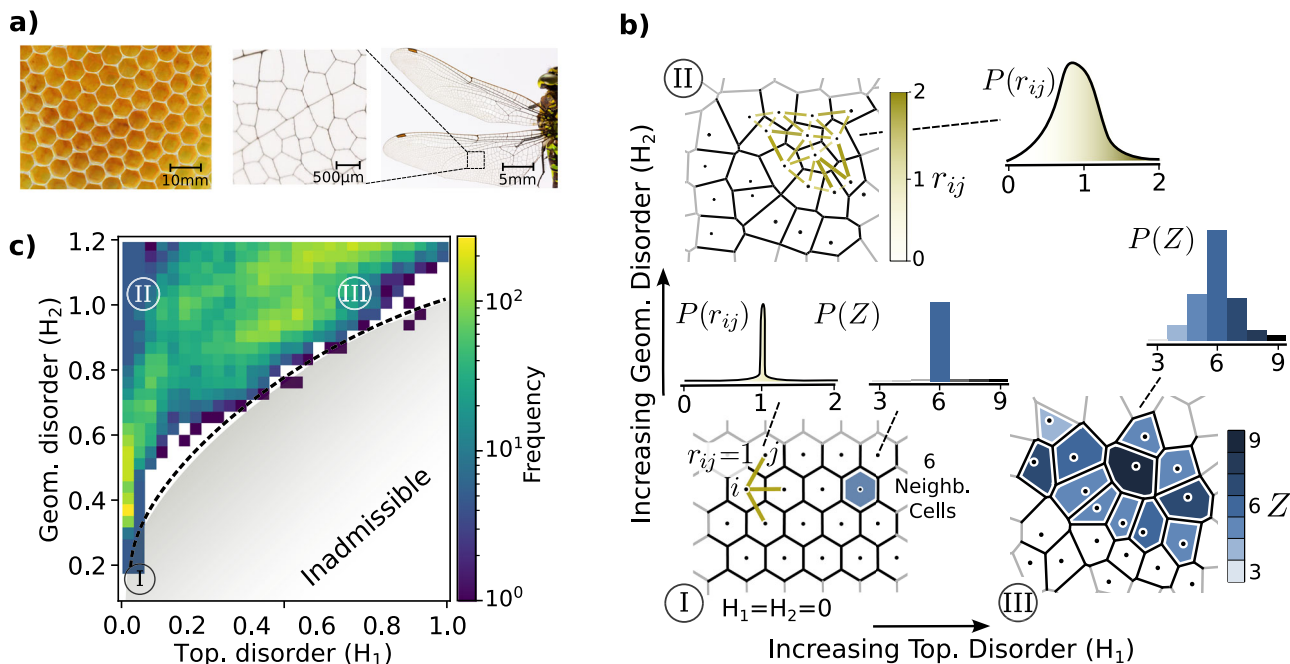


Fig. 1 Design space of disordered cellular materials. **a** Examples of a regular and an irregular cellular architecture in nature. Left: bee honeycomb (Source: Ante Hamersmit, CCO License), right: dragonfly wing (Source: Pixabay, CCO License). **b** I: Perfectly regular honeycomb. II: Interpretation of geometrical disorder in terms of the distribution of internuclei distances. III: Interpretation of topological disorder in terms of the distribution of local coordination numbers (number of cell neighbors). **c** Frequency histogram in disorder space for our full dataset.

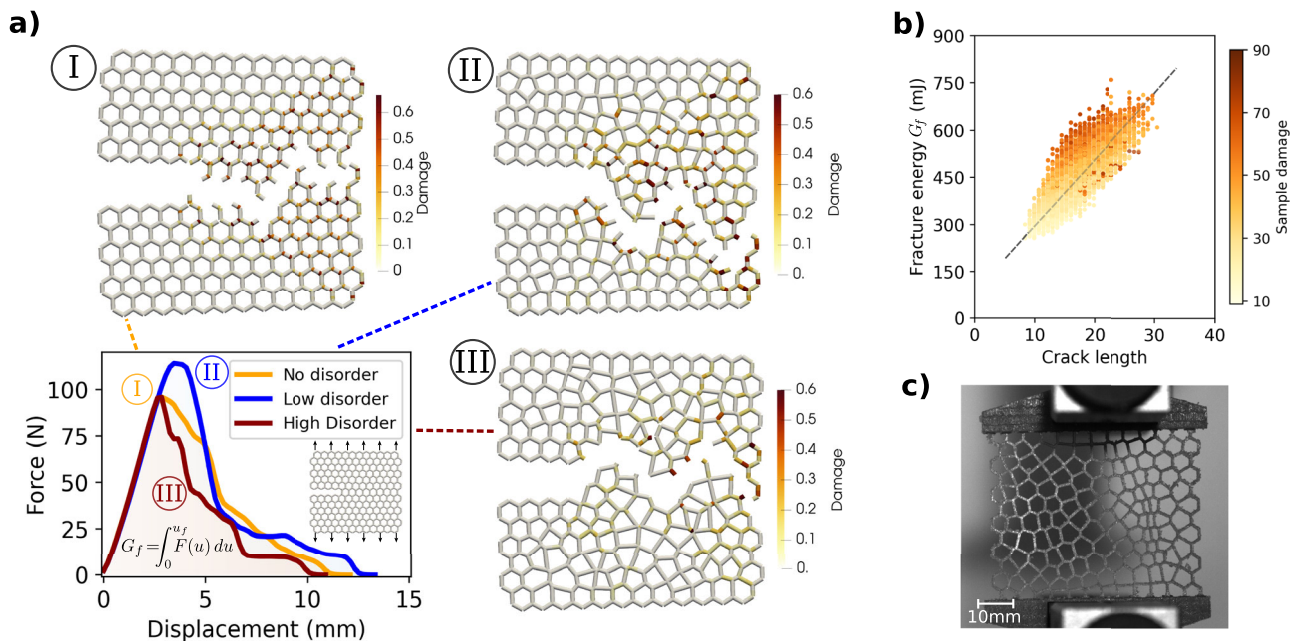


Fig. 2 Fracture mechanics of disordered cellular materials. **a** Force-displacement curves and corresponding fracture patterns, resulting from simulations of three specimens, using the elastoplastic beam finite element model and different levels of disorder: I) perfectly regular, II) low disorder, III) high disorder. **b** Fracture energy vs. crack length (number of broken edges). **c** Experimental setup for validation, using an unnotched irregular 3D-printed specimen.

The base material is described by a 1D model featuring rate-dependent damage and plastic hardening. The model is calibrated and validated against tensile experiments of samples that are additively manufactured out of a mixture of acrylic photopolymer and thermoplastic elastomer using a Connex polyjet printer (Fig. 2c). The details of the material model can be found in “Methods”, while its calibration and validation is presented in Supplementary Note 2.

In the case of quasi-regular networks ($H_2 < 0.5$), we observe oscillatory fracture patterns (e.g., Fig. 2a - I/II), verifying similar observations in earlier studies¹⁴, with cracks propagating continuously from the crack tip. As disorder increases, cracks may, in some cases, appear a few cells away from the crack tip and later coalesce, indicating a form of crack bridging¹, while, in other cases, smaller cracks disconnected from the main crack may form. The force-displacement curves in Fig. 2a reveal that some

degree of disorder may be beneficial for obtaining higher fracture energy $G_f = \int_0^{u_f} F(u)du$ (i.e., the area under the curve), where u and F denote the boundary displacement and net force, respectively. Interestingly, the data from ref. ¹ also hint at the possibility of a beneficial effect of moderate disorder, but the authors did not investigate this in detail. Figure 2b plots G_f for all networks in the dataset. As reported in prior studies of other materials (e.g., ref. ⁴³), we find that the value of G_f correlates with the length of the crack path (i.e., the number of broken edges). This is expected since most of the absorbed energy is derived from the release of elastic energy previously stored in beams that will eventually fracture. Several other mechanisms, including the accumulation of damage and plasticity in beams that will eventually not reach the point of fracture, are also at play but do not significantly affect the value of G_f .

Machine learning model. For each of the simulated samples, we save the updated graph connectivity every time a beam is broken, ignoring any isolated broken beams away from the crack path. This results in a sequence of as many graphs as the number of (non-isolated) broken edges. This set of sequences of graphs constitutes our training data. As shown in Fig. 3, the machine learning model takes as input the current configuration at step t (i.e., t broken edges), along with a hidden state encapsulating the history from previous steps. It outputs a fracture probability for all intact beams (network edges), along with a new hidden state. At any given time, the crack path is simply the set of edges that represent the difference in connectivity between the original graph ($t = 0$) and the current graph (t). The model features three fundamental components. The spatial component (Fig. 3a) accounts for the effect of the neighborhood of a crack tip on the stress concentration/redistribution and, hence, the crack advance. It is implemented as a custom message-passing graph convolution⁴⁴. The temporal component (Fig. 3b) incorporates any pertinent history effects of the process and is furnished by a gated recurrent unit⁴⁵. Finally, the inner-product decoder (Fig. 3c) is the mathematical machinery used to translate the

previous components into a probability of edge breaking. We briefly discuss each of the components in the following, yet a more detailed formulation can be found in Supplementary Note 3.

The purpose of the spatial message-passing convolution is to learn the relation between the topological and geometrical features encoded in the nodes and edges and the probability of an intact edge breaking next. Figure 3a shows the details of the graph convolution employed, which is based on the general framework of ref. ⁴⁴. We operate on the dual graph (Voronoi nuclei represent nodes of the dual graph, and edges between Voronoi cells correspond to the relevant dual edges⁴⁶), on which we define appropriate nodal features \mathbf{v}_i (cell volume, anisotropy vector, circularity, coordination number, geometrical order parameter) and edge features \mathbf{a}_{ij} (edge orientation, edge length). We refer to Supplementary Note 3 for a detailed mathematical definition of each feature. The nodal features are updated by a message-passing operation as follows: for each node i , we form a message \mathbf{m}_{ij} originating from a neighboring node j by passing the nodes' features $\mathbf{v}_i, \mathbf{v}_j$, their positions $\mathbf{x}_i, \mathbf{x}_j$ and the edge features \mathbf{a}_{ij} through a learnable function, here a fully connected neural network. This operation is summarized as $\mathbf{m}_{ij} = \phi_e(\mathbf{v}_i, \mathbf{v}_j, \mathbf{x}_i, \mathbf{x}_j, \mathbf{a}_{ij})$. The messages from the set of neighbors of i , i.e., $\mathcal{N}(i)$ are aggregated through a simple summation $\mathbf{m}_i = \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}$. A second neural network takes as input this aggregated message as well as the original node feature \mathbf{v}_i to produce the new node feature $\mathbf{v}'_i = \phi_v(\mathbf{v}_i, \mathbf{m}_i)$ for node i . This whole operation constitutes one convolutional layer and results in updating all nodal features by accumulating topological and geometrical information from neighbors one hop away. We find that six spatial convolutions produce the most accurate results. This mirrors the idea of a fracture process zone of a similar topological distance where stress redistribution occurs¹. Upon applying these convolutions, each node i now holds the final nodal features or embeddings $\mathbf{z} = \{\mathbf{z}_i\}_{i \leq N}$.

Since the fracture process can include history effects due to damage and plasticity, we also incorporate a temporal aspect

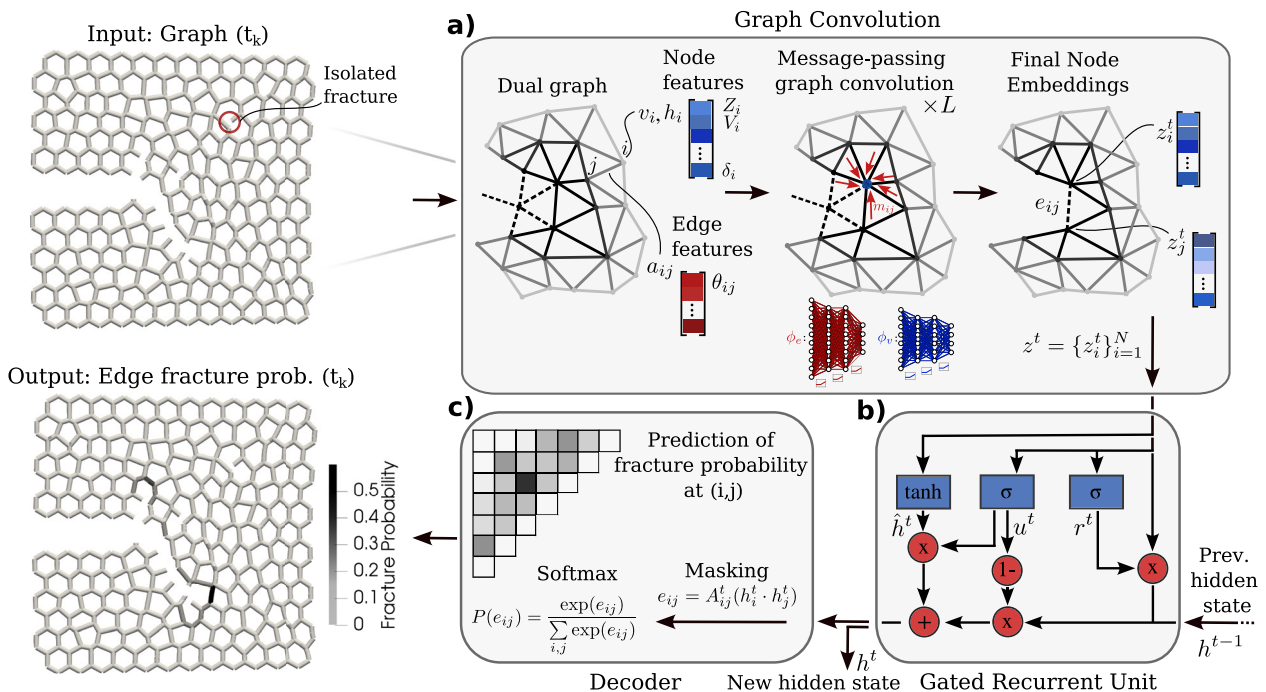


Fig. 3 Architecture of the spatiotemporal graph neural network. a Message-passing convolution, **b** Gated recurrent Unit, **c** Inner-product decoder. For better illustration, only the graph near the crack tip is shown.

into the model. We introduce a gated recurrent unit (GRU)⁴⁵, which is a modern and more efficient version of the long short-term memory network (LSTM)⁴⁷, a widely used recurrent neural network architecture. The underlying idea is to combine the node embeddings \mathbf{z}^t derived from the spatial convolution at a given step t , as described above, with a hidden embedding \mathbf{h}^{t-1} of the same dimensionality, which is continuously updated and incorporates the memory of all previous steps. As shown in Fig. 3b, the GRU incorporates an update gate $\mathbf{u}_t = \sigma(W_u[\mathbf{z}^t, \mathbf{h}^{t-1}] + \mathbf{b}_u)$ used to retain useful information from the node embeddings of the previous step, and a reset gate $\mathbf{r}_t = \sigma(W_r[\mathbf{z}^t, \mathbf{h}^{t-1}] + \mathbf{b}_r)$ used to forget the embedding information of the previous step, which is no longer relevant for future predictions. Note that σ , W , and \mathbf{b} denote the sigmoid activation function, weights, and biases of these gates, respectively, while \mathbf{h}^0 is a collection of zero vectors (embeddings), indicating the absence of memory at the first step. With the help of a few additional numerical manipulations, the next hidden state \mathbf{h}^t is computed, as described in detail in Supplementary Note 3. Overall, the GRU computes the state at step t through the hidden state at step $t-1$ and the current state at step t . Note that similar ideas of combining graph convolutions with recurrent architectures have been recently used in traffic forecasting⁴⁸.

To compute the likeliest edge to break at step t , we first pass the current hidden state \mathbf{h}^t through an inner-product decoder⁴⁹. This produces a matrix with components $\mathbf{h}_i^t \cdot \mathbf{h}_j^t$. Clearly, only existing edges are eligible to be broken; hence, we use the current adjacency matrix as a mask (via element-wise multiplication) to compute $e_{ij}^t = A_{ij}^t(\mathbf{h}_i^t \cdot \mathbf{h}_j^t)$, where no summation over identical indices is implied. For the sake of efficiency, we further assume that the crack advances by breakage of edges adjacent to the cells of the current path, although, as discussed above, in some occasions, edges may break up to a couple of cells away and then coalesce. In these cases, the order in which network edges are assumed to fracture deviates slightly from the ground truth. Finally, a softmax layer translates the value e_{ij}^t corresponding to each edge into a probability of breaking, $p(e_{ij}^t)$. The training of the model amounts to minimizing the binary cross entropy $\mathcal{L} = -1/N \sum_{i=1}^n \hat{p}_{ij}^t \log[p(e_{ij}^t)] + (1 - \hat{p}_{ij}^t) \cdot \log[1 - p(e_{ij}^t)]$, where $\hat{p}_{ij}^t = 1$ for the true edge that breaks at step t and 0 for all other edges. Additional details regarding hyperparameters, data augmentation, and training parameters are discussed in “Methods”.

Prediction accuracy. Once trained, the model predicts the crack advance at arbitrary lengths through an incremental approach, similar to ref. 18. To evaluate the accuracy of the trained model, we plot the ratio of correctly predicted edges (e.g., edges that were predicted to break and indeed broke) over the sum of correctly and incorrectly predicted edges for the entire fracture path (Fig. 4a). The average accuracy is similar for the training and test sets (both above 90%), indicating that no overfitting has taken place. Restricting this calculation to the first t steps (i.e., the first t broken beams of a fracture path) and repeating it for increasing t allows us to compute the timewise evolution of the accuracy. Figure 4b shows the statistics (mean \pm std. deviation) of this t -step accuracy across multiple predictions in the training and test sets. As expected, the accuracy decreases with increasing steps (i.e., increasing the number of broken beams) as it becomes more and more likely for the prediction to diverge from the correct fracture path. This represents a limitation of the current model in evaluating very long paths, which could be addressed with a more sophisticated temporal model. The inset of Fig. 6a shows the accuracy as a function of the geometrical disorder H_2 , which

highlights that the accuracy decreases as regularity increases. This is anticipated since, in the limit of perfect honeycombs, fracture may propagate upward or downward with equal probability due to symmetry. This is also evident in Fig. 4c, which compares model predictions and ground truth fracture patterns for samples randomly drawn from the training and test set at different values of disorder. For example, in the upper left plot (corresponding to low disorder), the model predicts a fracture pattern that is a mirror of the ground truth. As disorder increases, predictions generally improve. It is finally worth noting that the use of the softmax layer for the prediction of the next edge to break naturally provides an estimate of the confidence of the model’s prediction. The closer to 1 the value of the evaluated edge after the softmax layer, the more accurate the individual prediction is for a given increment.

Speed and memory efficiency. To gauge the performance of the framework in terms of speed, we measure the prediction time for 1000 different networks, sampled across the entire dataset, and compute the mean and standard deviation. For the same networks, we also calculate the statistics of the computation time of the highly optimized FEM solver. As shown in Fig. 5, the proposed graph learning framework delivers almost 2 orders of magnitude faster predictions than FEM. It is worth noting that this comes with a substantial memory footprint during training, which stems from expensive neighborhood aggregation operations^{50,51}. Fortunately, the nature of the problem allows us to keep the memory consumption constant despite increasing network size. This is due to the concept of a K-dominant region around the crack tip^{1,6}, outside of which the stresses decay rapidly. This implies that the prediction of the next edge to fail only depends on the subgraph covering a finite region around the instantaneous crack tip, while the rest of the graph can be ignored, allowing us to efficiently handle large graphs. If, for 3D extensions of the framework, that finite subgraph still leads to substantial memory consumption, then solutions can be sought in a combination of techniques, including using compressed⁵² or historical embeddings⁵³, decoupling neighborhood aggregation from prediction^{54,55}, adopting subgraph^{56,57} training, and using more scalable architectures relying on reversible connections⁵⁸ or anisotropic convolutions⁵⁹. All the above adaptations have the potential to reduce the framework’s memory footprint.

Optimizing the fracture toughness. We leverage our trained model toward designing cellular architectures with maximal fracture toughness. The latter is interpreted here as the fracture energy, which is widely used in similar studies^{39,60,61}. This interpretation of toughness overcomes insufficiencies of the LFM interpretation and the need to consider lattices where the region of K_I dominance is much larger than the cell size. Moreover, we can take advantage of the observed correlation between the fracture energy and the length of the crack path (Fig. 2b) and tackle the proxy problem of maximizing the latter.

We particularly consider two optimization problems. First, we focus on discovering specific Voronoi networks (i.e., spatial arrangements of nuclei) that exhibit the longest fracture paths. Since we can rapidly evaluate fracture patterns using the trained model, we opt for a sampling-based optimization. To do so, we sequentially perturb an initial arrangement of nuclei using a simple Markov Chain Monte Carlo (MCMC) scheme. A given perturbation is accepted with a probability that depends on the effected change of length of the fracture path, as predicted by the trained model. The algorithm is given in more detail in “Methods”. We carry out this optimization procedure starting from 20 initial arrangements and report the best optima. These

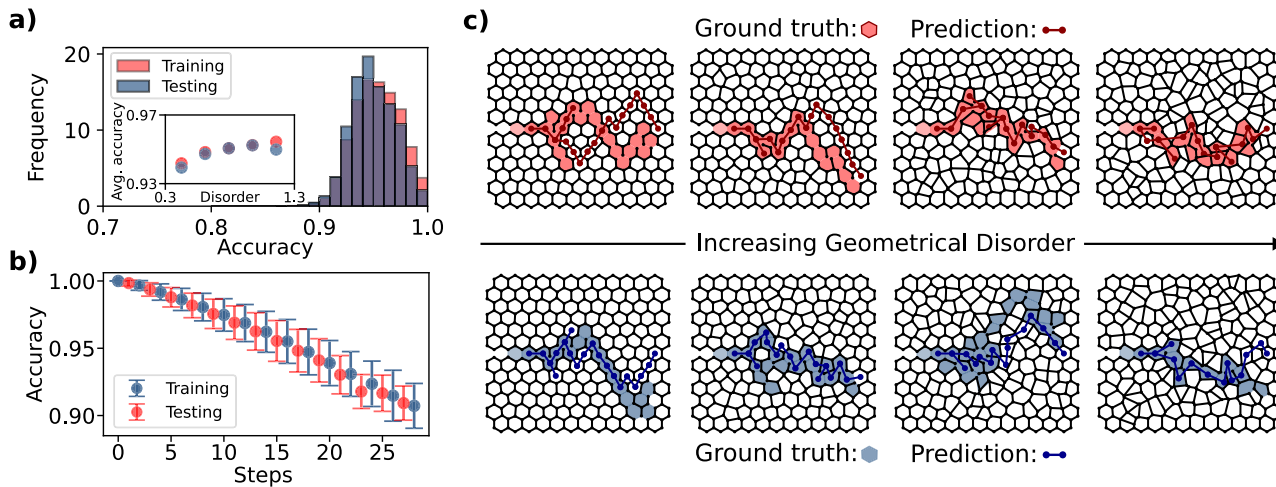


Fig. 4 Trained graph neural network. **a** Accuracy histogram for predictions on the training ($n = 45,000$ samples) and test dataset ($n = 5000$ samples) for the entire fracture path. Inset: Average accuracy vs. geometrical disorder. **b** Evolution of the accuracy of the prediction on the training and test dataset as a function of the number of steps (i.e., number of broken beams). Error bars represent ± 1 standard deviation. **c** Randomly sampled architectures for a range of different values of geometrical disorder (upper row: training set, lower row: test set). Ground truth (light filled nuclei—broken primal edges) is compared with model predictions (dark dual edges).

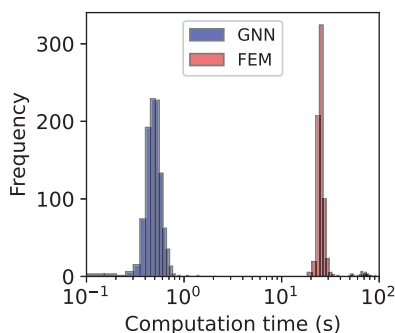


Fig. 5 Speed performance of the framework. Comparison between the time in seconds for predicting the fracture paths using FEM and the proposed graph framework ($n = 1000$ independent experiments).

are shown in Fig. 6a in terms of the topological crack length and the associated fracture energy, evaluated upon the end of optimization by FEM analysis. The corresponding optimizers (Voronoi networks) and the fracture patterns are shown in Fig. 6b. Interestingly, the optimal networks exhibit relatively diffuse crack patterns, which include crack branching. Finally, we report the evolution of the predicted crack length and disorder metrics throughout the optimization process in Fig. 6c and d, respectively.

Second, ensemble optimization allows us to look for the generators of the ensemble (λ_i , see “Methods”) that produce networks exhibiting, on average maximal crack path lengths and hence maximal fracture energies. To this end, we adapt the statistical physics-inspired design engine set forth in ref. ⁶² to our problem. The major advantage of this method is that it incorporates knowledge about the configuration space. We start with an initial value of the generators and incrementally update them in an optimal manner. The exact algorithm is detailed in “Methods”. Figure 7a shows the evolution of the probability density of the geometrical and topological disorder of the networks until convergence. Note that the ensemble optimal networks lie in a region of the disorder space characterized by higher regularity (lower H_1, H_2 values) compared to the previous individually optimized networks. In particular, we find that the

optimal topological disorder approaches zero, i.e., topological defects, on average, do not increase the fracture energy. Finally, Fig. 7b shows the evolution of λ_i , while Fig. 7c shows the average crack length of the evolving ensemble until convergence. We are able to achieve a 20% increase in this average length.

Discussion

We have developed a design methodology for irregular architected cellular materials, which represents a paradigm shift from conventional periodic designs. By utilizing two effective statistical metrics that represent topological and geometrical disorders, we have introduced a new approach for parameterizing the design of such materials. By moving these statistical descriptors from the dual to the primal space, the design space can be expanded beyond Voronoi-based architectures. We have focused on the challenging task of optimizing their fracture energy, with a wide range of applications in materials science and engineering, including the design of tough structural materials and aerospace components⁶³, biomedical implants⁶⁴, and electronics⁶⁵. Our machine learning framework, which allows for the rapid prediction of crack patterns in these cellular media, leverages their natural description as graphs and results in a learned mapping from the space of local topological and geometrical features at the vicinity of a crack tip to its incremental crack advance. The trained model relies on a number of assumptions, including mode-I loading as well as a specific notch configuration and beam description. However, the framework itself is applicable under a wider range of conditions (e.g., mode-II loading or Timoshenko beams) as long as there exists a pronounced notch leading to stress concentrations. It will be interesting to investigate its generalization capabilities under these modified assumptions. We have also shown how the framework can be used for the design of optimal architectures that maximize fracture energy. Unlike in prior approaches, disorder in cellular materials is explicitly used to optimize nonlinear properties beyond the simple engineering of defects in an otherwise perfectly periodic architecture. Finally, the framework can prove useful in designing the fracture path, e.g., steering it away from specific parts of the domain, or it may be utilized toward optimizing combined mechanical properties beyond fracture. Overall, this study can pave the way for a new

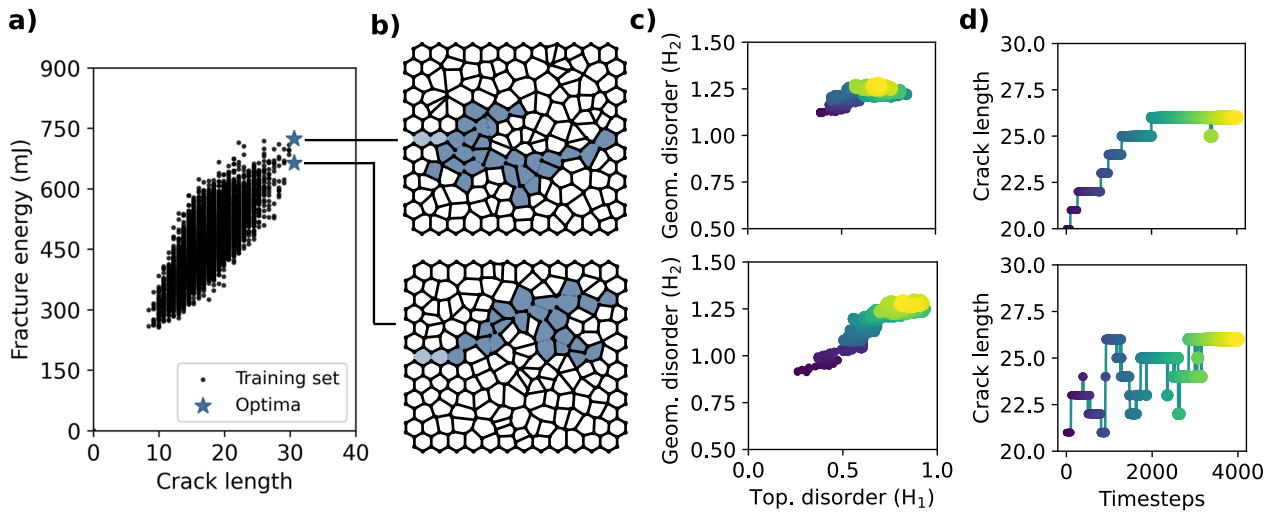


Fig. 6 Sample optimization. **a** Fracture energy vs. topological crack path length for the training set (·) and two examples of optimal points obtained through MCMC (★) along with **b** the corresponding optimizers (Voronoi networks). **c** Evolution of the topological and geometrical disorder for each of the networks throughout the optimization. **d** Evolution of the predicted crack length throughout the optimization. The color map indicates optimization steps from beginning (●) to end (●).

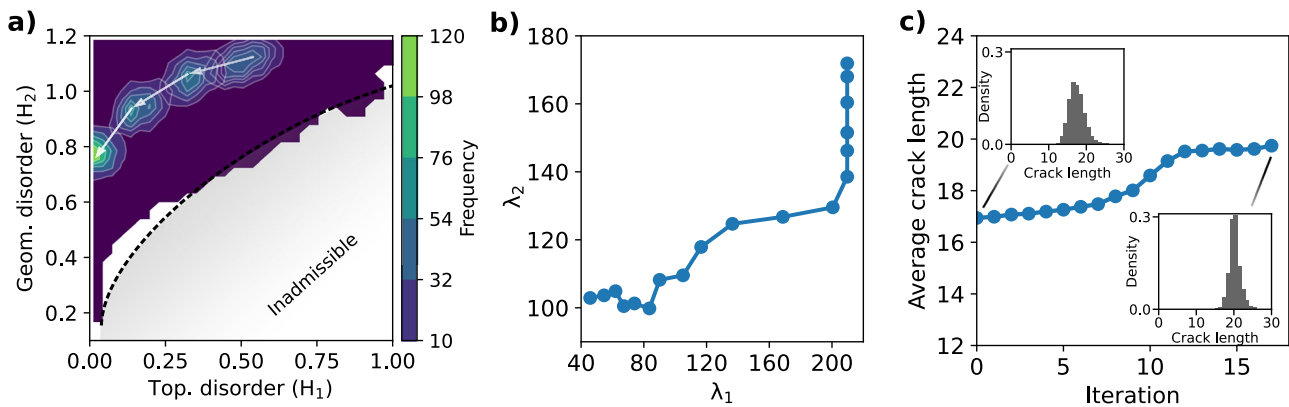


Fig. 7 Ensemble optimization. **a** Evolution of the probability density of topological and geometrical disorder during optimization. **b** Evolution of the generators (Lagrange multipliers) controlling the Voronoi networks. **c** Average crack length vs. iteration number.

class of architected disordered materials, especially upon its extension to 3D.

Methods

Generation of random architectures. The disordered architectures in this study have been generated as follows. We consider a statistical ensemble \mathcal{G}_n of Voronoi networks composed of n nuclei. The ensemble is constructed by maximizing the entropy subject to a normalization constraint and two additional constraints which serve to control the degree of disorder. This is mathematically stated as

$$\begin{aligned} \max_P & - \sum_{G \in \mathcal{G}_n} P(G) \log P(G), \\ \text{s.t.} & \sum_G P(G) H_i(G) = \hat{H}_i(G), \quad i = 1, 2, \\ & \sum_G P(G) = 1, \end{aligned} \tag{1}$$

where G denotes the Voronoi network or graph, $P(G)$ is the probability of its occurrence in the ensemble, $H_1 = \langle Z^2 \rangle - \langle Z \rangle^2$ is a measure of the topological disorder of a network (i.e., the variance of the local coordination number Z), and $H_2 = \langle r_{ij}^2 \rangle - \langle r_{ij} \rangle^2$ is a measure of its geometrical disorder (i.e., the variance of the internuclei distance r_{ij}). The solution of Eq. (1) gives the well-known Boltzmann distribution:

$$P(G) = \frac{1}{\mathcal{Z}} \exp \left(- \sum_i \lambda_i H_i(G) \right), \tag{2}$$

where \mathcal{Z} denotes the partition function. In practice, instead of enforcing specific values of \hat{H}_i directly, we consider a range of corresponding λ_n , then sample from the corresponding distributions above using MCMC⁶⁶, and inspect the resulting H_i .

This construction provides greater control than the typical construction of random Voronoi lattices based only on the minimum internuclei distance (see, e.g., ref. 40). Note that sampling one Voronoi network requires 5000–10,000 MCMC iterations, depending on the λ_i -values (i.e., the degree of disorder). During each iteration, the nuclei positions are perturbed, a new Voronoi tessellation is computed, the resulting “energy” of the system, $\sum \lambda_i H_i$, is calculated, and the perturbed positions are accepted or rejected based on standard MCMC criteria. The entire process takes, on average, 120s in our unoptimized Python code running on an Intel i7 1.30GHz processor.

Fracture simulations. We adopt an Euler–Bernoulli corotational formulation for the beam kinematics in our cellular architectures⁶⁷. We postulate the existence of an energy potential $\mathcal{W}(\epsilon, \epsilon_p)$ as well as a dissipation potential $\phi^*(\dot{\epsilon}_p)$, where ϵ is the longitudinal strain and ϵ_p is the plastic longitudinal strain. Restricting our attention to slender beams, we adopt a fiber-type formulation⁶⁸, in which the strain contributions are uncoupled. In order to capture the mechanical behavior seen in experiments, we adopt the following form for the potentials, based on a plastic strain driven damage-law (d), a power law-based plastic hardening energy, and a rate-dependent dissipation potential:

$$\mathcal{W} = \frac{1}{2} (1 - d) E (\epsilon - \epsilon^p)^2 + \frac{C}{\kappa + 1} |\epsilon^p|^{\kappa+1}, \tag{3}$$

$$d = 1 - \exp(-s|\epsilon^p|), \tag{4}$$

$$\phi^*(\dot{\epsilon}^p) = \sigma_0 |\dot{\epsilon}^p| + \tau_0 \frac{\dot{\epsilon}_0}{m+1} \left| \frac{\dot{\epsilon}^p}{\dot{\epsilon}_0} \right|^{m+1}, \tag{5}$$

depending on Young’s modulus E , hardening parameters C and κ , damage

parameter s , and strength parameters σ_0 , τ_0 , and m . Additional details about the model formulation, its calibration and validation against experiments are given in Supplementary Notes 1 and 2.

Graph neural network model. Our graph neural network depicted in Fig. 3a includes fully connected edge and node networks, each having three layers and feature rectified linear units as activation layers. The overall graph network has three layers, a hidden nodal dimension of 256, and a latent nodal dimension of 32. A relevant hyperparameter study can be found in Supplementary Note 4. Residual layers are introduced in the graph network to avoid the known problem of oversmoothing⁶⁹. Dropout layers with a dropout probability of 0.4 are used to avoid overfitting. For the purpose of data augmentation, the domain and the associated fracture pattern are flipped vertically, effectively doubling the size of the dataset. During training, we monitor the model performance on the test set, comprised of 10% of the total data, after each epoch. We employ the ADAM optimizer⁷⁰, with a learning rate of 0.00005. We perform early stopping when the testing accuracy metric stops decreasing for six consecutive epochs to avoid overfitting. The model is implemented in the open-source library Pytorch-Geometric⁷¹.

Fracture path optimization—sample. We adopt a sequential optimization algorithm furnished by a Markov Chain Monte Carlo approach. The algorithm takes the following steps: (1) take N nuclei and distribute them either randomly on the domain or such that they correspond to a desired initial value of topological and geometrical disorder. (2) Construct the Voronoi graph and compute associated node/edge features. (3) Evaluate the trained machine learning model to predict the fracture path length L . (4) Move the nucleus of a randomly chosen Voronoi cell by a small random displacement. (5) Recompute (locally) the Voronoi tessellation, update the graph features, and reevaluate the fracture path length L' . If $L' > L$, accept the move. If $L' \leq L$ only accept with a probability $e^{-(L'-L)/T}$, where T is a temperature-like hyperparameter, otherwise do not move the nucleus. (6) Check for convergence in the overall improvement of L . (7) Go to step (4).

Fracture path optimization—ensemble. For the ensemble average optimization, we adopt a statistical physics-inspired optimization scheme. The fundamental idea is to compute the evolution of the Lagrange multipliers λ_i toward the direction of increasing the probability of finding the system in states with above-average values of a desired average quantity. This quantity here is the length of the crack path, $L(G)$, given a Voronoi graph G . Following ref. ⁶², we desire

$$\dot{P}(G|\lambda_i) = P(G|\lambda_i)[L(G) - \langle L(G) \rangle], \quad (6)$$

where $\langle \cdot \rangle$ denotes averaging over the configuration space weighted by P_{λ_i} . This translates into

$$\dot{\lambda}_i = (\partial_{\lambda_i} \log(P) \partial_{\lambda_j} \log(P))^{-1} [L(G) - \langle L(G) \rangle] \partial_{\lambda_j} \log(P). \quad (7)$$

Using Eq. (2), the above equation further transforms into

$$\dot{\lambda}_i = (H_i(G) H_j(G))^{-1} [L(G) H_j(G)]. \quad (8)$$

Averaging over the configuration space is computed by sampling 2000 configurations in parallel. For each, the crack path can be rapidly evaluated using the trained graph model.

Data availability

The source data for all figures, as well as the data required to train the machine learning model, are available in the ETH Research Collection (<https://doi.org/10.3929/ethz-b-000608722>)⁷².

Code availability

The code for the machine learning model developed in this study is available from GitHub (<https://github.com/kkarapiperis/gnn-fracture>).

Received: 20 January 2023; Accepted: 23 May 2023;

Published online: 02 June 2023

References

- Christodoulou, I. & Tan, P. J. Crack initiation and fracture toughness of random Voronoi honeycombs. *Eng. Fract. Mech.* **104**, 140–161 (2013).
- Manno, R., Gao, W. & Benedetti, I. Engineering the crack path in lattice cellular materials through bio-inspired micro-structural alterations. *Extrem. Mech. Lett.* **26**, 8–17 (2019).
- Fleck, N. A., Deshpande, V. S. & Ashby, M. F. Micro-architected materials: past, present and future. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **466**, 2495–2516 (2010).
- Meza, L. R. et al. Resilient 3D hierarchical architected metamaterials. *Proc. Natl Acad. Sci. USA* **112**, 11502–11507 (2015).
- du Plessis, A. et al. Properties and applications of additively manufactured metallic cellular materials: a review. *Prog. Mater. Sci.* **125**, 100918 (2022).
- O'Masta, M. R., Dong, L., St-Pierre, L., Wadley, H. N. G. & Deshpande, V. S. The fracture toughness of octet-truss lattices. *J. Mech. Phys. Solids* **98**, 271–289 (2017).
- Fleck, N. A. & Qiu, X. The damage tolerance of elastic-brittle, two-dimensional isotropic lattices. *J. Mech. Phys. Solids* **55**, 562–588 (2007).
- Shaikeea, A. J. D., Cui, H., O'Masta, M., Zheng, X. R. & Deshpande, V. S. The toughness of mechanical metamaterials. *Nat. Mater.* **21**, 297–304 (2022).
- Betego'n, C. & Hancock, J. W. Two-parameter characterization of elastic-plastic crack-tip fields. *J. Appl. Mech.* **58**, 104–110 (1991).
- Tankasala, H. C., Deshpande, V. S. & Fleck, N. A. 2013 Koiter Medal Paper: crack-tip fields and toughness of two-dimensional elastoplastic lattices. *J. Appl. Mech.* **82**, 091004 (2015).
- Hsieh, M.-T., Deshpande, V. S. & Valdevit, L. A versatile numerical approach for calculating the fracture toughness and R-curves of cellular materials. *J. Mech. Phys. Solids* **138**, 103925 (2020).
- Romijn, N. E. R. & Fleck, N. A. The fracture toughness of planar lattices: imperfection sensitivity. *J. Mech. Phys. Solids* **55**, 2538–2564 (2007).
- Curtin, W. A. & Scher, H. Brittle fracture in disordered materials: a spring network model. *J. Mater. Res.* **5**, 535–553 (1990).
- Schmidt, I. & Fleck, N. A. Ductile fracture of two-dimensional cellular structures—dedicated to Prof. Dr.-Ing. D. Gross on the occasion of his 60th birthday. *Int. J. Fract.* **111**, 327–342 (2001).
- Phani, A. S. & Fleck, N. A. Elastic boundary layers in two-dimensional isotropic lattices. *J. Appl. Mech.* **75**, 021020 (2008).
- Mateos, A. J., Huang, W., Zhang, Y.-W. & Greer, J. R. Discrete-continuum duality of architected materials: failure, flaws, and fracture. *Adv. Funct. Mater.* **29**, 1806772 (2019).
- Ambati, M., Gerasimov, T. & De Lorenzis, L. A review on phase-field models of brittle fracture and a new fast hybrid formulation. *Comput. Mech.* **55**, 383–405 (2015).
- Hsu, Y.-C., Yu, C.-H. & Buehler, M. J. Using deep learning to predict fracture patterns in crystalline solids. *Matter* **3**, 197–211 (2020).
- Abraham, F. F., Brodbeck, D., Rafey, R. A. & Rudge, W. E. Instability dynamics of fracture: a computer simulation investigation. *Phys. Rev. Lett.* **73**, 272 (1994).
- Desai, J., Allaire, G. & Jouve, F. Topology optimization of structures undergoing brittle fracture. *J. Comput. Phys.* **458**, 111048 (2022).
- Kumar, S. & Kochmann, D. M. In *Current Trends and Open Problems in Computational Mechanics*, 275–285 (Springer, 2022).
- Moore, B. A. et al. Predictive modeling of dynamic fracture growth in brittle materials with machine learning. *Comput. Mater. Sci.* **148**, 46–53 (2018).
- Schwarzer, M. et al. Learning to fail: predicting fracture evolution in brittle material models using recurrent graph convolutional neural networks. *Comput. Mater. Sci.* **162**, 322–332 (2019).
- Perera, R., Guzzetti, D. & Agrawal, V. Graph neural networks for simulating crack coalescence and propagation in brittle materials. *Comput. Methods Appl. Mech. Eng.* **395**, 115021 (2022).
- Pierson, K., Rahman, A. & Spear, A. D. Predicting microstructure-sensitive fatigue-crack path in 3D using a machine learning framework. *JOM* **71**, 2680–2694 (2019).
- Kim, M., Winovich, N., Lin, G. & Jeong, W. Peri-net: analysis of crack patterns using deep neural networks. *J. Peridyn. Nonlocal Model.* **1**, 131–142 (2019).
- Hsu, Y.-C., Yu, C.-H. & Buehler, M. J. Tuning mechanical properties in polycrystalline solids using a deep generative framework. *Adv. Eng. Mater.* **23**, 2001339 (2021).
- Wang, Y. et al. Stressnet-deep learning to predict stress with fracture propagation in brittle materials. *NPJ Mater. Degrad.* **5**, 1–10 (2021).
- Lew, A. J., Yu, C.-H., Hsu, Y.-C. & Buehler, M. J. Deep learning model to predict fracture mechanisms of graphene. *NPJ 2D Mater. Appl.* **5**, 1–8 (2021).
- Buehler, M. J. Modeling atomistic dynamic fracture mechanisms using a progressive transformer diffusion model. *J. Appl. Mech.* **89**, 121009 (2022).
- Dewapriya, M. A. N., Rajapakse, R. K. N. D. & Dias, W. P. S. Characterizing fracture stress of defective graphene samples using shallow and deep artificial neural networks. *Carbon* **163**, 425–440 (2020).
- Font-Clos, F. et al. Predicting the failure of two-dimensional silica glasses. *Nat. Commun.* **13**, 1–11 (2022).
- Sen, D. & Buehler, M. J. Structural hierarchies define toughness and defect-tolerance despite simple and mechanically inferior brittle building blocks. *Sci. Rep.* **1**, 1–9 (2011).
- Li, T., Chen, Y. & Wang, L. Enhanced fracture toughness in architected interpenetrating phase composites by 3D printing. *Compos. Sci. Technol.* **167**, 251–259 (2018).

35. Tankasala, H. C. & Fleck, N. A. The crack growth resistance of an elastoplastic lattice. *Int. J. Solids Struct.* **188–189**, 233–243 (2020).
36. Reiser, J. & Hartmaier, A. Elucidating the dual role of grain boundaries as dislocation sources and obstacles and its impact on toughness and brittle-to-ductile transition. *Sci. Rep.* **10**, 1–18 (2020).
37. Lew, A. J. & Buehler, M. J. A deep learning augmented genetic algorithm approach to polycrystalline 2D material fracture discovery and design. *Appl. Phys. Rev.* **8**, 041414 (2021).
38. Gu, G. X., Wettermark, S. & Buehler, M. J. Algorithm-driven design of fracture resistant composite materials realized through additive manufacturing. *Addit. Manuf.* **17**, 47–54 (2017).
39. Shu, X. et al. Toughness enhancement of honeycomb lattice structures through heterogeneous design. *Mater. Des.* **217**, 110604 (2022).
40. Zhu, H. X., Hobdell, J. R. & Windle, A. H. Effects of cell irregularity on the elastic properties of 2D Voronoi honeycombs. *J. Mech. Phys. Solids* **49**, 857–870 (2001).
41. Luan, S., Chen, E. & Gaitanaros, S. Energy-based fracture mechanics of brittle lattice materials. *J. Mech. Phys. Solids* **169**, 105093 (2022).
42. ASTM International. *Standard Test Methods for Measurement of Fracture Toughness* (ASTM, 2019).
43. Gustafsson, A., Wallin, M., Khayyeri, H. & Isaksson, H. Crack propagation in cortical bone is affected by the characteristics of the cement line: a parameter study using an XFEM interface damage model. *Biomech. Model. Mechanobiol.* **18**, 1247–1261 (2019).
44. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. *PMLR* **70**, 1263–1272 (2017).
45. Cho, K., Van Merriënboer, B., Bahdanau, D. & Bengio, Y. On the properties of neural machine translation: Encoder-Decoder approaches. In *Proc. SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation 2014*, 103–111 (2014).
46. Aurenhammer, F. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.* **23**, 345–405 (1991).
47. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
48. Zhao, L. et al. T-GCN: a temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **21**, 3848–3858 (2019).
49. Kipf, T. N. & Welling, M. Variational graph auto-encoders. In *Proc. NIPS Workshop Bayesian Deep Learn.*, 2016, 1–3 (2016).
50. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *Proc. International Conference on Learning Representations 2017 (ICLR, 2017)*.
51. Cai, T. et al. GraphNorm: a principled approach to accelerating graph neural network training. *PMLR* **139**, 1204–1215 (2021).
52. Liu, Z. et al. Exact: scalable graph neural networks training via extreme activation compression. In *Proc. International Conference on Learning Representations 2022 (ICLR, 2022)*.
53. Fey, M., Lenssen, J. E., Weichert, F. & Leskovec, J. Gnnautoscale: scalable and expressive graph neural networks via historical embeddings. *PMLR* **139**, 3294–3304 (2021).
54. Wu, F. et al. Simplifying graph convolutional networks. *PMLR* **97**, 6861–6871 (2019).
55. Huang, Q., He, H., Singh, A., Lim, S.-N. & Benson, A. R. Combining label propagation and simple models out-performs graph neural networks. In *Proc. International Conference on Learning Representations, 2021 (ICLR, 2021)*.
56. Chen, J., Ma, T. & Xiao, C. FastGCN: fast learning with graph convolutional networks via importance sampling. In *Proc. International Conference on Learning Representations, 2018 (ICLR, 2018)*.
57. Chiang, W.-L. et al. Cluster-GCN: an efficient algorithm for training deep and large graph convolutional networks. In *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 257–266 (2019).
58. Li, G., Müller, M., Ghanem, B. & Koltun, V. Training graph neural networks with 1000 layers. *PMLR* **139**, 6437–6449 (2021).
59. Tailor, S. A., Opolka, F., Lio, P. & Lane, N. D. Do we need anisotropic graph neural networks? In *Proc. International Conference on Learning Representations 2022 (ICLR, 2022)*.
60. Mayer, G. Rigid biological systems as models for synthetic composites. *Science* **310**, 1144–1147 (2005).
61. Dimas, L. S., Bratzel, G. H., Eylon, I. & Buehler, M. J. Tough composites inspired by mineralized natural materials: computation, 3D printing, and testing. *Adv. Funct. Mater.* **23**, 4629–4638 (2013).
62. Miskin, M. Z., Khaira, G., de Pablo, J. J. & Jaeger, H. M. Turning statistical physics models into materials design engines. *Proc. Natl Acad. Sci. USA* **113**, 34–39 (2016).
63. Schaedler, T. A. & Carter, W. B. Architected cellular materials. *Annu. Rev. Mater. Res.* **46**, 187–210 (2016).
64. Gibson, L. J., Ashby, M. F. & Harley, B. A. *Cellular Materials in Nature and Medicine* (Cambridge University Press, 2010).
65. Qiu, L., He, Z. & Li, D. Multifunctional cellular materials based on 2D nanomaterials: prospects and challenges. *Adv. Mater.* **30**, 1704850 (2018).
66. Robert, C. P., Casella, G. & Casella, G. *Monte Carlo Statistical Methods*, Vol. 2 (Springer, 1999).
67. Phlipot, G. P. & Kochmann, D. M. A quasicontinuum theory for the nonlinear mechanical response of general periodic truss lattices. *J. Mech. Phys. Solids* **124**, 758–780 (2019).
68. Spacone, E., Filippou, F. C. & Taucer, F. F. Fibre beam–column model for non-linear analysis of R/C frames: Part I. Formulation. *Earthq. Eng. Struct. Dyn.* **25**, 711–725 (1996).
69. Chen, D. et al. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proc. AAAI Conference on Artificial Intelligence*, Vol. 34, 3438–3445 (2020).
70. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *Proc. International Conference on Learning Representations 2015 (ICLR, 2015)*.
71. Fey, M. & Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *Proc. ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019).
72. Karapiperis, K. & Kochmann, D. M. Dataset: Prediction and control of fracture paths in disordered architected materials using graph neural networks (ETH Research Collection, 2023).

Acknowledgements

K.K. acknowledges the support from a Marie-Sklodowska Curie Postdoctoral Fellowship under Grant Agreement No. 101024077. The authors also acknowledge fruitful discussions with Jan-Hendrik Bastek.

Author contributions

K.K. and D.M.K. conceptualized the project. K.K. and D.M.K. designed the methodology. K.K. wrote the code, carried out the experiments and simulations, and trained the machine learning model. K.K. and D.M.K. procured funding. D.M.K. supervised the project. The initial version of the manuscript was written by K.K.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s44172-023-00085-0>.

Correspondence and requests for materials should be addressed to Dennis M. Kochmann.

Peer review information *Communications Engineering* thanks the anonymous reviewers for their contribution to the peer review of this work. Primary handling editors: Miranda Vinay and Rosamund Daw.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023