



Encoding and exploring latent design space of optimal material structures via a VAE-LSTM model

Andrew J. Lew^{a,b}, Markus J. Buehler^{a,c,*}

^a Laboratory for Atomistic and Molecular Mechanics (LAMM), Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, United States

^b Department of Chemistry, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, United States

^c Center for Computational Science and Engineering, Schwarzman College of Computing, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, United States

ARTICLE INFO

Keywords:
 Mechanics
 Design
 Machine learning
 3D printing
 Autoencoder
 Topology optimization
 Structural design
 Optimization
 Deep learning
 AI

ABSTRACT

Variational autoencoders (VAE) are machine learning models that can extract low dimensional representations of data from datasets of high complexity and volume. Importantly, they can be used for generative purposes to reconstruct complex data, such as images, from a low dimensional encoding of only a few variables. Long short-term memory (LSTM) neural networks are well suited to learning logical trajectory relationships within datasets. Using these two models in concert, we develop a VAE-LSTM approach to learn a classic mechanical materials design problem. Here, we focus on the compliance optimization of cantilever design, using a VAE to encode cantilever structures into a 2D latent space and a LSTM to learn trajectories in that latent space corresponding to the optimization process. Ultimately, we are able to clearly visualize the space of cantilever design, generate new design with extremely low density beyond the original dataset, and obtain optimal cantilever structures inspired by nature. We also demonstrate how the resulting designs can be manufactured using 3D printing, offering a rapid pathway from concept to prototype. The method we developed here can be generalized to other image-based datasets encapsulating changes from multiple factors. The ability offered by our approach to interpret complex behavior, via representations in simplified space, has great potential for application in the intelligent design and manufacturing of materials structure problems.

1. Introduction

Autoencoding models are unsupervised learning algorithms that learn to map an input to an identical output[1] through a set of recognition weights and generative weights[2]. By incorporating a low dimensional bottleneck layer between recognition and generation, these neural network models yield representations of otherwise complex datasets in a simplified latent space suitable for data compression, visualization, and exploratory data analysis[3]. Unlike principal component analysis (PCA), autoencoders can achieve nonlinear dimensionality reduction, which allows a more general treatment of complex datasets with greater accuracy so long as requisites in computational power, dataset size, and initial weights are met[4]. Variational autoencoders (VAE)[5,6] introduce a probabilistic reparameterization of the encoded vector, acting to regularize the low dimensional latent space through the Kullback-Leibler divergence[7] such that points close in latent space yield correspondingly similar data when decoded. This

regularization helps enforce continuity and completeness of the latent space, which combats overfitting and is necessary for the model to act as a generative tool in obtaining logical new structures from arbitrary latent vectors.

Here we develop a method to probe materials structure optimization using a VAE model. Specifically, we do so with focus on the compliance minimization of a loaded cantilever as an example. By training the VAE on structures with various volume fractions at different stages of compliance optimization, we obtain low dimensional encodings for the general space of loaded cantilever structures. Subsequently, we can easily understand and explore the space of potential cantilevers via exploration of this low dimensional latent space. Using a long short-term memory (LSTM) neural network, which are suitable for time series predictions and datasets of prescribed ordering[8,9], to learn the motion of optimization in latent space allows us to generate new reasonable structures and replicate the optimization process for arbitrary proposed structures. The generative power of the combined VAE-LSTM model is

* Corresponding author.

E-mail address: mbuehler@mit.edu (M.J. Buehler).

demonstrated with the ability to obtain structures of volume fraction extrapolated beyond the training data, inspired from unique structural motifs such as the image of a leaf, and to finish design drafts of flawed proposed structures.

Artificial intelligence methods are quickly becoming an important tool in understanding the intimate connection between material structure and property[10], with recent success predicting stress and strain fields in composites[11,12], fracture mechanisms of 2D crystalline materials[13,14], and damage mechanics of bone tissue[15]. Concurrently, AI tools have aided in efforts to generate and grasp novel structures via the exploration and exploitation of vast design spaces[16], sonification based encodings of amino acid sequences[17,18] and sonification of fire [19]. While VAEs and LSTMs have been used in concert in the literature for complex systems such as noisy soft-sensor models[20], animations of human motion[21], and stock market price predictions[22], to our knowledge this is the first instance of such a VAE-LSTM model applied to materials structure optimization. Though here we focus specifically on the singular task of compliance minimization in developing the method, the approach has broader impact in treating analogous datasets that evolve spatiotemporally. The method allows us to walk the fine line of generating structures that are simultaneously in family with previous examples and of novel quality.

2. Results and discussion

The classic cantilever problem can be formulated as a rectangular space with the left end fixed in place and a downward load acting on the bottom right corner of the structure. These boundary conditions are provided, along with a prescribed volume fraction of the design space for the cantilever to fill, as input to an open-source topology optimization code[23]. Further details on this code are provided in the Materials and methods section. Using this code yields a series of images as the algorithm iteratively generates structures toward optimal minimum compliance under the provided conditions, as shown in Figure 1a. Specifically, we run the algorithm to generate 151 snapshots of progressive optimization for each of 48 volume fractions from 5% to 99%, to obtain a total dataset of 7,248 images.

This dataset represents the complex process of cantilever design as a function of volume fraction and degree of compliance optimization. We use these images, each sized to 512 by 256 pixels, to train a VAE model in order to obtain a low dimensional representation of the salient factors connecting structure to property. Specifically, we utilize a 6-layer encoder to obtain vector encodings in 2-dimensional latent space, which can be used to reconstruct the original images by an 8-layer decoder, as shown in Figure 1b. The dimensionality of the latent space is up to our choosing, and here we select to encode images down to two latent variables for ease of visualization in a 2D scatter plot. The reconstructed images are visually identical to the original images, indicating the model has successfully extracted two latent variables able to represent changes in the 512 by 256-pixel image space.

Three or more latent variables are likely going to decrease the degree of data compression, and allow for less lossy reconstructions. Taking the size of the latent space up to the limit of the size of the original data will allow for perfect reconstructions with the identity function, but this is of course the trivial case as there is no longer any data compression. Therefore, a good balance is necessary between a useful degree of dimensionality reduction and maintaining well reconstructed images. In the case studied here, the grayscale 512 by 256-pixel images are simple enough to be represented in 2D latent space without visible degradation to the reconstructed images, so increasing number of latent variables beyond two is unnecessary.

The 2D vector latent space representations of the original images are easily visualizable in a scatter plot, which is shown in Figure 2a. These points are colored by the volume fraction of the structure that each point in latent space corresponds to, confirming gradual regularity of the latent space without abrupt cliffs and valleys characteristic of overfitting. Accordingly, we can see the limited design space accessed over the topology optimization algorithm, with optimal structures lying on a curve bounding the left and upper regions of the space. Using the decoder module of our VAE model allows us to generate structures beyond those represented in the training data by simply providing the desired latent space coordinates as input. A grid of cantilever structures filling out the rest of the immediate latent space is shown in Figure 2b.

Beyond simply generating a volume of novel structures from latent

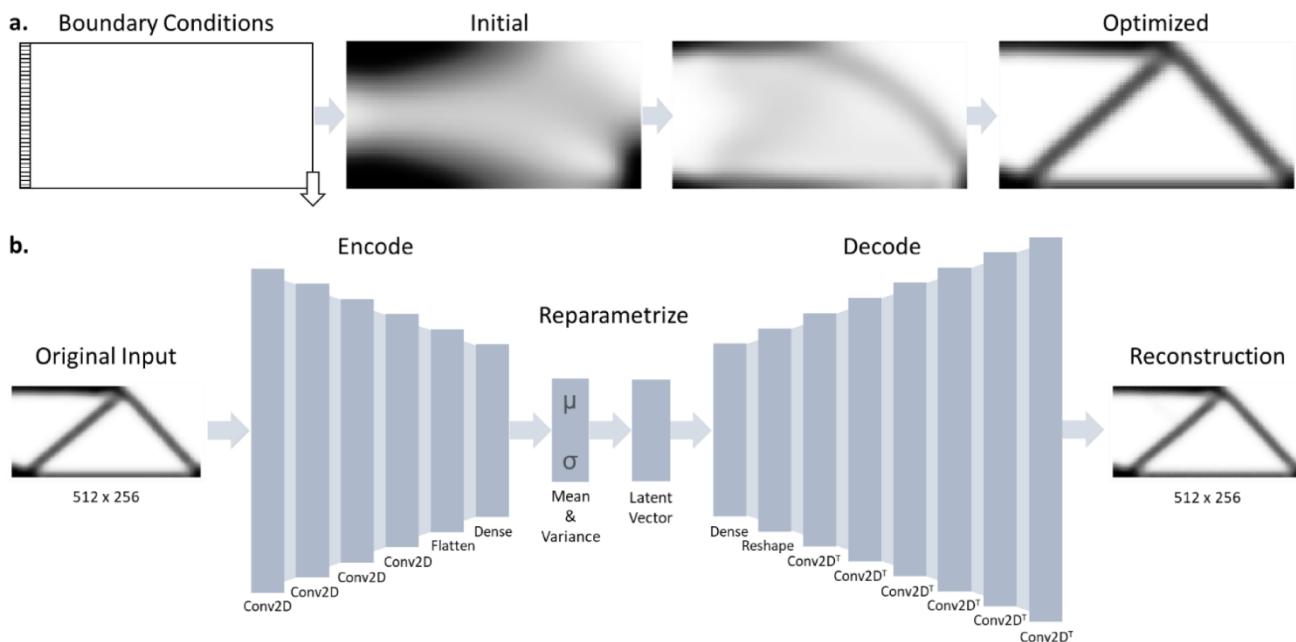


Fig. 1. Compressing topology optimization into low dimensional latent space with VAE. For a given volume fraction and boundary conditions, **a.** the TopOpt algorithm iteratively generates structures to achieve optimal compliance. We obtain a dataset of 7248 images, comprised of 48 volume fractions from 5% to 99%, over 151 snapshots of the optimization process. These images are used to train a **b.** variational autoencoder model, which reduces each image into a 2D latent vector that can recreate the original image after decoding.

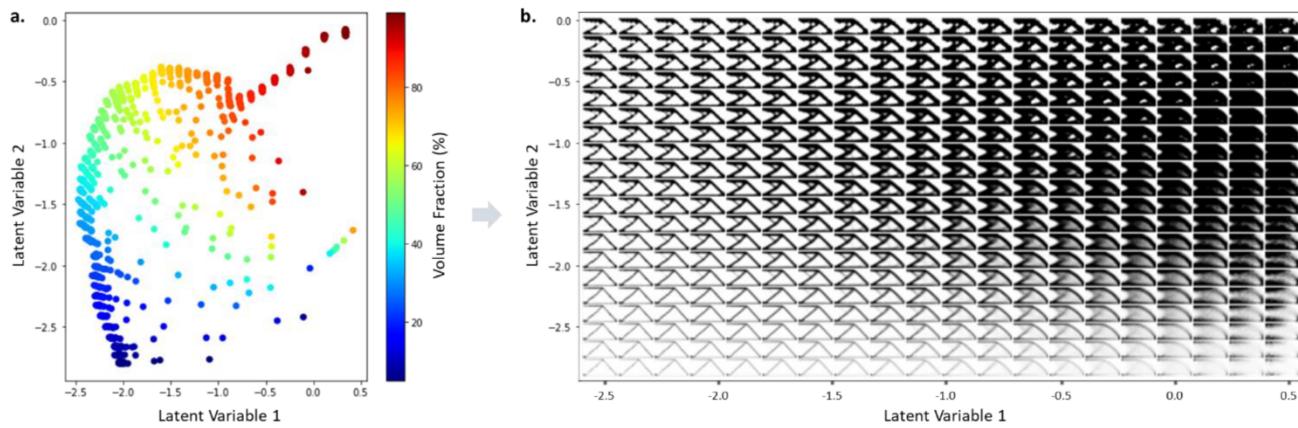


Fig. 2. Encoding and decoding structures with 2D latent space. The **a.** 2D latent variable encodings can be easily plotted and colored by volume fraction in order to visualize the design space probed by the topology optimization algorithm. We can use the decoder to **b.** visualize structures throughout the latent space, extrapolating beyond the training data to obtain structures in between known training points, in regions beyond the optimal curve inaccessible by the optimization algorithm, and extrapolating to potentially optimized solutions missed by the training dataset. Though it can generate a large diversity of structures, by itself the VAE model is unable to distinguish good solutions from bad solutions or understand the optimization process.

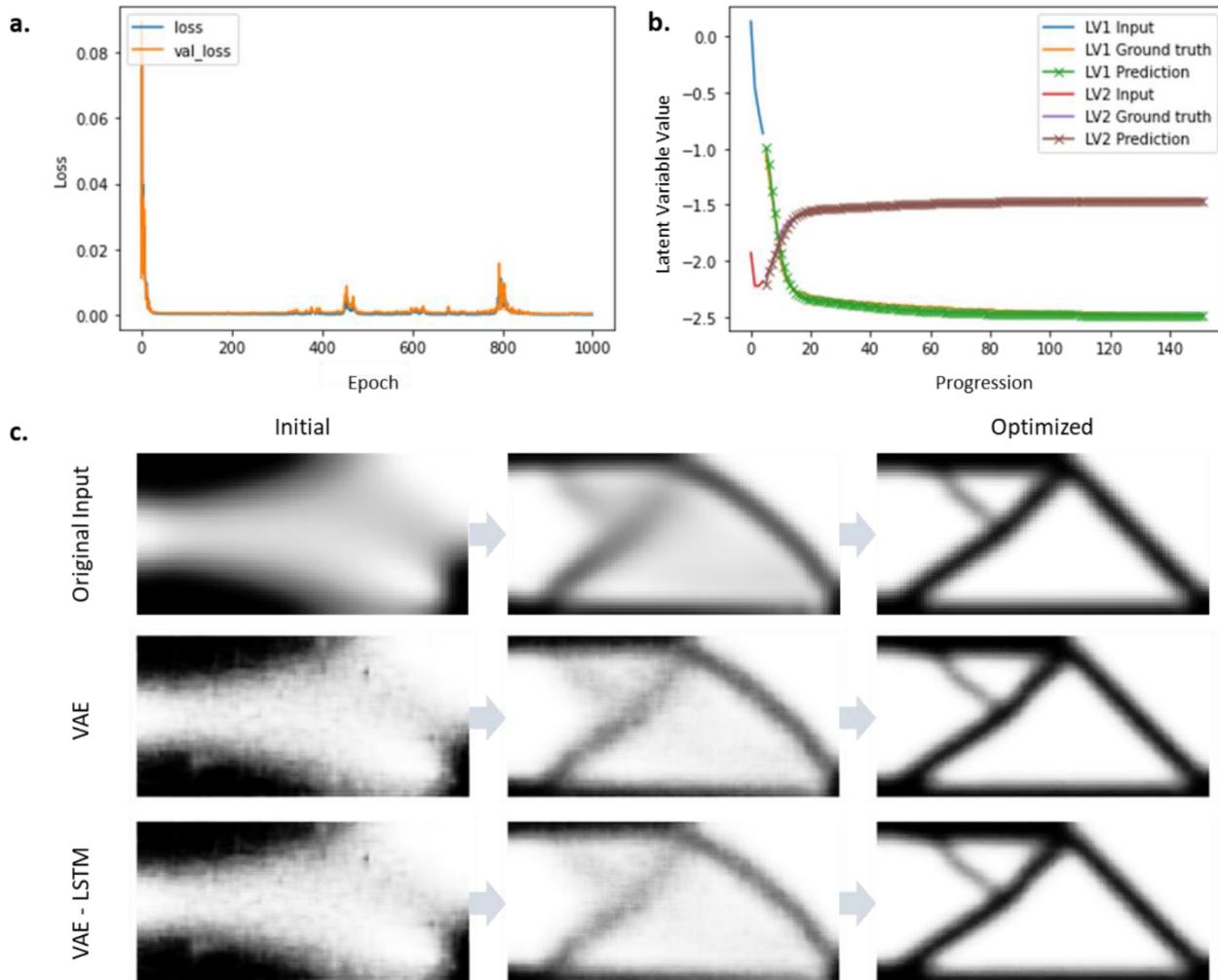


Fig. 3. Learning relationships between structures in latent space via LSTM model. We train an LSTM model on the progression of latent space variables over topology optimization iterations with **a.** low near-zero losses and good agreement between training and validation sets. Using a volume fraction of 33% as an example, **b.** the model can accurately predict the evolution of the two latent space variables over the optimization process with near perfect agreement between the predictions and ground truth. Decoding the latent space variables to reconstructed images, we see **c.** both the VAE and VAE-LSTM structures agree with the original images across the optimization process.

space, we use an LSTM model to learn logical trajectories through latent space that correspond to valid optimization routes. Over the course of training the model for 1000 epochs, in **Figure 3a** we obtain near zero losses with good agreement between training and validation sets, which indicates good predictive efficacy without overfitting. Further details of the LSTM model are provided in the Materials and Methods section.

The trained LSTM model is able to accurately predict the evolution of latent space coordinates, as indicated in **Figure 3b**. Using a volume fraction of 33% as an example, the quantitative predicted values of the two latent variables LV1 and LV2 almost exactly overlap the true optimization trajectory. Qualitatively, we compare the original images across the optimization process against the VAE reconstructed images and the images decoded from LSTM-predicted latent space coordinates in **Figure 3c** and find nearly identical structures. Full video of the VAE-LSTM predicted structures over the course of optimization are provided in the Supplementary Information Movie M1.

The combined VAE-LSTM model is then able to rapidly generate complex structures from simple inputs, and evolve them in accordance with the classical topology optimization algorithm. In this way, novel structures can be quickly accessed. For example, in **Figure 4a** we can select a point in latent space near the lower edge of the region mapped out by the original topology optimization algorithm, such as the latent variable encoding $[-1, -2.75]$. The VAE-LSTM model evolves that starting point to its corresponding optimized form to obtain a valid cantilever structure with even lower volume fraction than anything previously seen in the training data. In addition to this kind of extrapolation task,

we can use the model to take structural inspiration from disparate sources to use as a base for subsequent structure optimization. In **Figure 4b** we use a leaf as one such example structure, obtain its representation in latent space with the VAE encoder module, and use the VAE-LSTM model to yield a new optimized cantilever structure of intermediary volume fraction. The model can also check the validity of proposed structures and finish the optimization process for flawed proposals. For example, **Figure 4c** shows a misshapen cantilever which is corrected by the VAE-LSTM model to the properly optimized morphology. **Figure 4d** presents a proposed cantilever with unnecessary extra supports, and the model refines this structure by redistributing mass from the extraneous supports to reinforce the important struts. **Figure 4e** shows the latent space trajectories of these four examples, with the latent space encodings of the initial structures evolving in the direction of the arrows to rest on the curve of optimal solutions. Full video of how these structures evolve are provided in the Supplementary Information Figures S2-S5.

The resulting designs can be manufactured using 3D printing. **Figure 5** shows five prints reflecting the designs featured in Supplementary Information Movies M1-M5, in increasing volume fraction from top to bottom. These prototype structures can be used to verify proposed designs with physical characterization techniques. The rapid nature of additive manufacturing compared to traditional fabrication methods allows for experimental finalization of production parameters such as out-of-plane thickness or surface treatments prior to full scale fabrication.

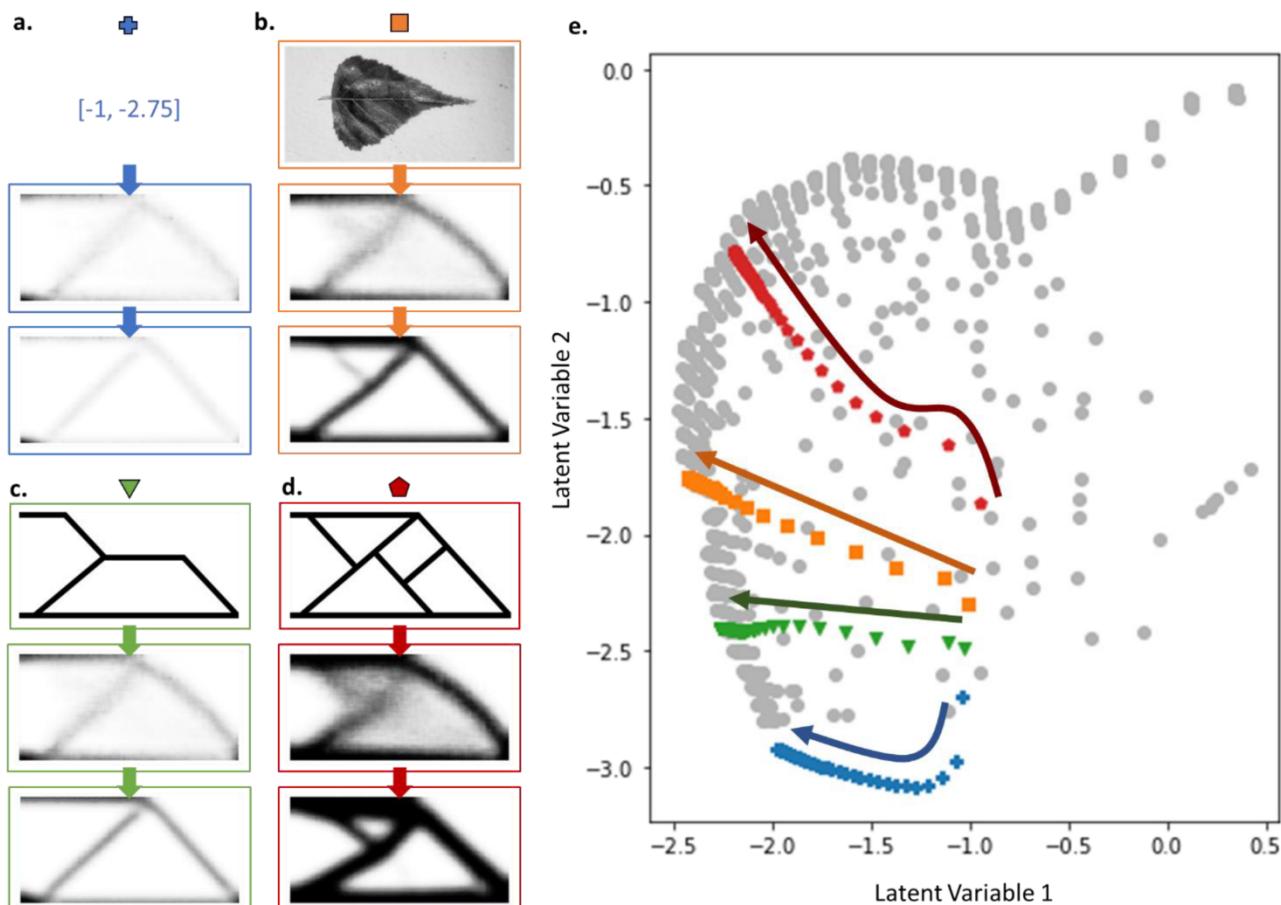


Fig. 4. Generating new optimized structures. Using the VAE-LSTM model, we can extrapolate to optimized structures with **a.** lower volume fraction than seen in the training data by directly probing latent space coordinates at the edge of the training data. We can also generate cantilevers **b.** inspired from other sources such as a leaf, by encoding it into latent space and running the model to complete structure optimization. Design drafts of **c.** misshapen cantilevers and **d.** over supported cantilevers can be presented to the VAE-LSTM model in order to obtain properly optimized final structures. The VAE-LSTM model is used to help correct and refine these proposed design drafts, respectively. **e.** The optimization paths in latent space of these four examples are shown, plotted by their respective marker symbols in a-d, illustrating that all structures converge on the optimal curve after repeated iterations.

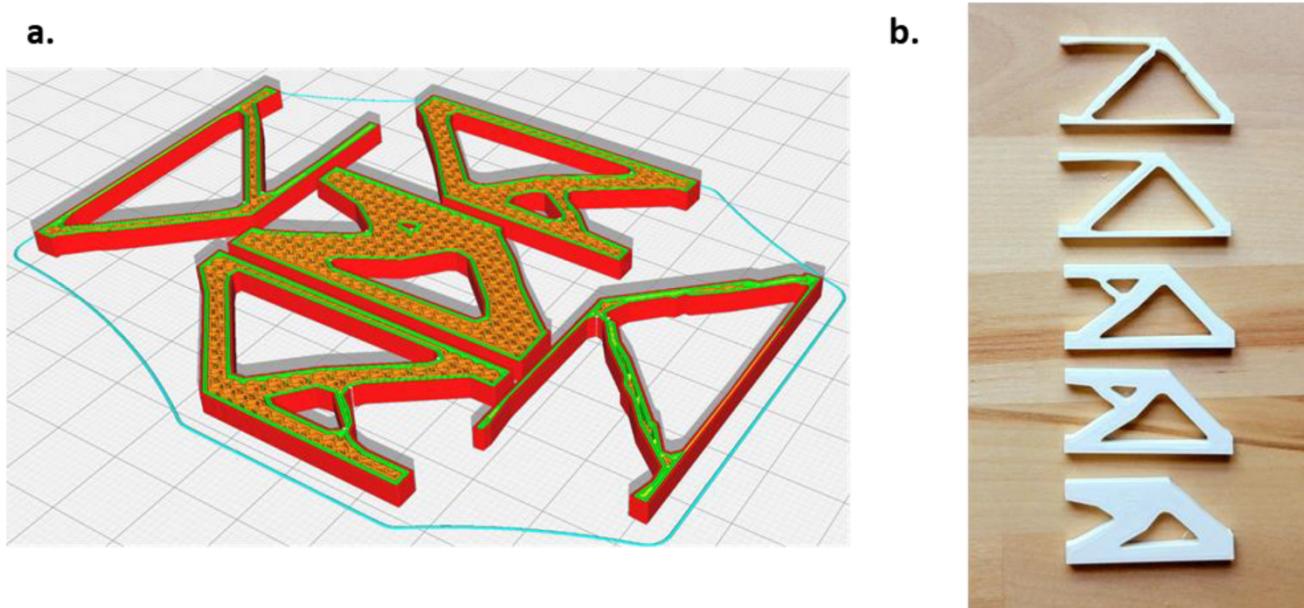


Fig. 5. Additive manufacturing results of 5 sample geometries generated by the model. Using 3D printing we generate physical prototypes of the resulting designs (**a**; sliced view of the samples, showing the internal gyroid structure chosen as internal fill; **b**; printed samples). The right panel shows 5 different optimized designs with increasing volume fraction from top to bottom. In order, the 5 designs reflect the structures obtained from low extrapolated volume fraction, misshapen, leaf-inspired, 33% volume fraction, and over supported cantilevers, respectively.

Moreover, since the deep learning model can efficiently generate stacks of images, these can be used to reconstruct 3D geometries based on an inverse tomography method (see Materials and methods for detail). [Figure 6](#) shows examples that clearly visualize the effect of changing latent variable values on the cantilever structure. By taking trajectories in latent space horizontally, vertically, and along the curve of all optimal structures, and stacking the progression of structure over these trajectories in the out-of-plane z-direction, we can turn motions from abstracted latent space into easily visualizable 3D models.

3. Conclusion

The use of a VAE-LSTM model allows us to reduce complex changes in structure to movements in easily understandable and interpretable 2D maps. A classic issue with VAE models is opacity in what the extracted latent variables represent, thus making it normally difficult to navigate latent space in a deliberate, meaningful way. The addition of a LSTM portion, trained on logical progressions of the original dataset, ameliorates this issue and allows for a truly predictive tool able to extrapolate beyond the original training data[24]. Such a model can be used to generate new structures in family to previous designs directly from latent space coordinates, or via inspiration from other structures, and act to finish design drafts if presented with a structure at a partial level of optimization. Our machine learning method may be used as a rapid screening tool to quickly gauge the validity of an architect or product engineer's design concepts prior to devoting the computational resources for a full finite element analysis or experimental resources for a physical mock up.

Importantly, the strategy of extracting a latent space with a VAE model followed by learning latent trajectories with a LSTM model is not inherently limited to a compliance optimization dataset or particular boundary conditions, as used in these examples introducing the method. A dataset of structures substituting volume fraction and compliance with grain size and hardness, for example, can be treated in an analogous manner. While the current study has been limited to a single set of boundary conditions, training against a larger dataset with a diversity of supports and loads or directly including these conditions as additional training labels will allow construction of a more general model. For

more complex datasets, a latent space of larger dimension may assist in maintaining predictive accuracy while avoiding overfitting. An immediately relevant application of our model is in use as a generative tool to obtain material structures with extraordinary properties, or to generate image stacks that can be reconstructed into 3D models and then manufactured using additive methods, as shown in [Figure 6](#). Supplementary Figure S1 shows a collection of images of the 3D printed sample shown in [Figure 6](#), taken from different angles.

While the current example uses 'optimization paths' as the dataset trajectories, the model as is could also be used to simply understand changes in a variable without an imposed ranking of what values are 'better'. In short, the VAE-LSTM architecture can act as a surrogate model to predict dynamic changes in structure, in general. We look forward to further work using this approach in better understanding connections and relationships between material structures and their properties, such as the progression of material microstructures in organisms or in designing materials with extreme mechanical properties like ultra-high hardness, toughness, or other target properties.

4. Materials and methods

4.1. Topology optimization algorithm

We use here an adaptation of an open-source Python implementation [23] of the efficient 88-line MATLAB code for topology optimization [25] based on the classic 99-line educational code[26] from the literature. In brief, the system is initialized with a rectangular design domain discretized by square finite elements. Compliance minimization of statically loaded structures is based upon the power-law approach, wherein material properties are assumed constant within discrete elements and modelled as the "relative material density raised to some power times the material properties of solid material"[26]. The program then incorporates Optimality Criteria methods[27], mesh-independency filtering[28], and finite element code to arrive at the optimized structures.

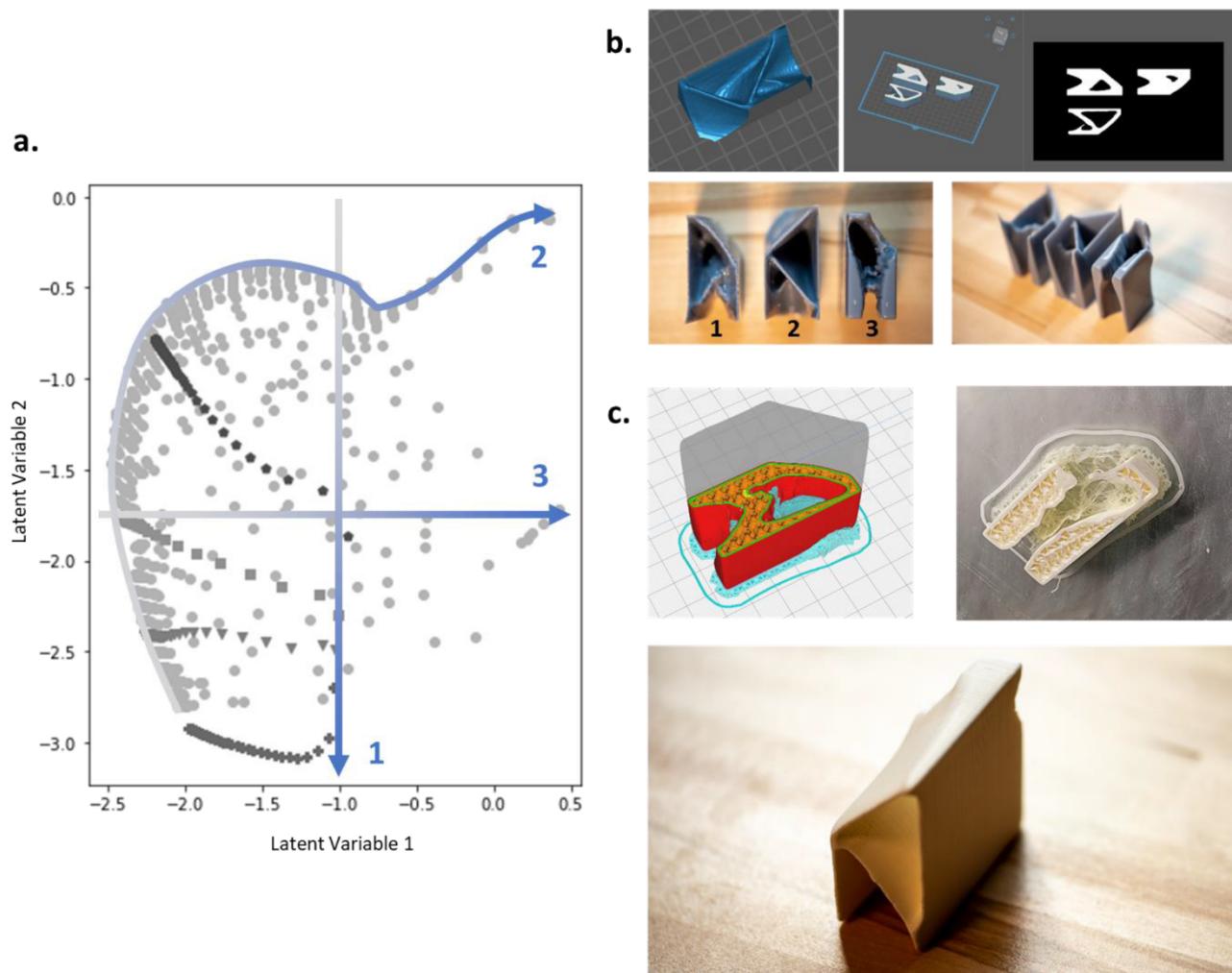


Fig. 6. Additive manufacturing results of 3D models generated from image stacks derived from movements in latent space. Various examples of movements in latent space are depicted, with **a**. three paths corresponding to a sweep in the vertical direction, along the curve of optimal solutions, and a sweep in the horizontal direction, labelled 1, 2, and 3 respectively. Cantilever structures are then decoded from points along these trajectories, stacked in the z-direction, and generated using **b**. resin-based printing (labelled according to the trajectories in panel a) and **c**. one example based on FDM realized using multimaterial printing using PLA and PVA as dissolvable support material. Supplementary Figure S1 shows a collection of images of the 3D printed sample, taken from different angles.

4.2. LSTM model to predict latent space trajectories

We train the LSTM network by first representing trajectories in latent space as lists of 151 sequential latent variable values, each corresponding to a step in the original topology optimization progression. We then split these lists into two sets, the initial four values to be used as an input set and the subsequent 147 values to be used as the output set. We then use the output set as a label to supervise the training of latent space coordinate predictions using the Adam[29] optimizer and mean squared error loss. 75% of the input–output sets are immediately used for training, with the remaining 25% reserved for validation. Implementation is in Python with the TensorFlow package[30]. The specific architecture of the LSTM model is provided in Table 1.

We use a Stacked LSTM comprised of four total LSTM layers as detailed in Table 1. This deep model was chosen to increase capacity beyond a single standard LSTM layer, potentially allowing the hidden state at each level to operate at different timescales[31]. This structure is intended to allow the model to capture differing rates of growth or shrinkage in different trusses over the cantilever optimization, for instance. Multidimensional and Grid LSTMs increase depth of the model by stacking cells not just on top of each other but also along the depth dimension. However, the values in the memory vectors of Multidimensional LSTMs can grow combinatorially with dimension[32]. While

Table 1
LSTM architecture.

Layer	Output Shape	Parameters
LSTM	(None, None, 40)	6880
LSTM	(None, 40)	12960
Dense	(None, 80)	3280
Activation	(None, 80)	0
RepeatVector	(None, 147, 80)	0
LSTM	(None, 147, 40)	19360
LSTM	(None, 147, 40)	12960
TimeDistributed	(None, 147, 2)	82
Activation	(None, 147, 2)	0

Grid LSTMs address the combinatorial growth problem for better efficiency[33], the added complexity of these two models beyond the Stacked LSTM implemented here are not necessary for our current system of study. It may, however, be useful to explore such types of neural networks in other applications or variations of the method reported here.

4.3. Visualizing the effect of latent variables and motions in latent space to prepare stacked images for manufacturing

The dependency of cantilever structure on the two latent variables can be directly probed by providing points of interest to the decoder module. To visualize the effect of Latent Variable 1 (as described in Figure 2) we first hold Latent Variable 2 constant at a value of -1.75, near the middle of the latent space scatter plot. Then, we horizontally sweep values for Latent Variable 1 from -2.5 to 0.5 in increments of 0.1 to generate a series of cantilever structure images. Similarly, to probe the effect of Latent Variable 2, we hold Latent Variable 1 constant at a value of -1 and vertically sweep values of Latent Variable 2 from 0 to -3.5 in increments of -0.1. We additionally follow the points along the upper-left bounding curve in latent space to visualize the progression of optimal solutions in order from lowest to highest volume fraction. These three cases are each rendered in 3D by stacking the structures obtained along these latent space trajectories. 3D printed models are then fabricated to clearly represent the changes in abstracted latent variable space as variations in the spatial z-axis.

4.4. Additive manufacturing of resulting designs

Based on the models generated, we develop 3D printed models by building an STL model. Different methods to generate 3D models were used. The simple geometries shown in Figure 5 were created based on 3D Builder (Microsoft), using images as basis to generate a 3D profile map. The STL model is used for 3D printing. The final image of each case is used by creating a contour map based on pixel color intensity (black=high, white=low).

For the complex geometries shown in Figure 6, 3D models of image sequences are generated using Dragonfly (2021.1.0.977, ORS Inc.). The images are loaded and the three RGB channels stitched into a single channel through their max intensity, then rendered into a mesh. Smoothing operations are applied to reduce pixilation in the final mesh that is used for 3D printing. The meshes may require additional processing, done here using Autodesk Meshmixer using the “Make Solid” algorithm. Subsequent remeshing and/or smoothing operations are applied to render a solid, watertight, smooth and printable geometry.

The resulting STL files are then sliced into g-code using Cura (Version 4.10.0), and printed using white PLA filament in a Fuse Deposition Molding Ultimaker S3 printer. Complex geometries are printed using PVA support material in multi-material printing mode. We also use resin SLA printing, where files are sliced using Chitobox and then printed using a Photon Mono X LCD printer, with grey 405 nm UV cross-linkable resin.

Authors' Contributions

M.J.B. conceived the idea and developed the VAE-LSTM model. A.J.L. implemented the analyses and analyzed the data with M.J.B. M.J.B. supervised the project. A.J.L. wrote the manuscript with M.J.B. M.J.B. 3D printed the models and conducted imaging.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This material is based upon work supported by the NSF GRFP under Grant No. 1122374. We acknowledge support by the Office of Naval Research (N000141612333 and N000141912375), AFOSR-MURI (FA9550-15-1-0514) and the Army Research Office (W911NF1920098). Related support from the IBM-MIT AI lab, MIT Quest, and Google Cloud Computing, is acknowledged.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.finmec.2021.100054.

References

- [1] G. Dong, G. Liao, H. Liu, G. Kuang, A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images, *IEEE Geosci. Remote Sens. Mag.* 6 (2018) 44–68.
- [2] G. Hinton, R. Zemel, Autoencoders, Minimum Description Length and Helmholtz Free Energy, *Adv. Neural Inf. Process. Syst.* 6 (1994) 3–10.
- [3] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, *AIChE J.* 37 (1991) 233–243.
- [4] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science* 313 (2006) 504–507.
- [5] D.P. Kingma, D.J. Rezende, S. Mohamed, M. Welling, Semi-supervised Learning with Deep Generative Models, in: *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.
- [6] D.J. Rezende, S. Mohamed, D. Wierstra, Stochastic Backpropagation and Approximate Inference in Deep Generative Models, *Proc. 31st Int. Conf. Mach. Learn.* 32 (2014) 1278–1286.
- [7] Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. arXiv Prepr. 1312.6114 (2013).
- [8] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (1997) 1735–1780.
- [9] F.A. Gers, J. Urgen Schmidhuber, F. Cummins, Learning to Forget: Continual Prediction with LSTM, *Neural Comput.* 12 (2000) 2451–2471.
- [10] K. Guo, Z. Yang, C.-H. Yu, M.J. Buehler, Artificial intelligence and machine learning in design of mechanical materials, *Mater. Horizons* 8 (2021) 1153–1172.
- [11] Z. Yang, C.-H. Yu, M.J. Buehler, Deep learning model to predict complex stress and strain fields in hierarchical composites, *Sci. Adv.* 7 (2021) eabd7416.
- [12] Z. Yang, C.H. Yu, K. Guo, M.J. Buehler, End-to-end deep learning method to predict complete strain and stress tensors for complex hierarchical composite microstructures, *J. Mech. Phys. Solids* 154 (2021), 104506.
- [13] Y.C. Hsu, C.H. Yu, M.J. Buehler, Using Deep Learning to Predict Fracture Patterns in Crystalline Solids, *Matter* 3 (2020) 197–211.
- [14] A.J. Lew, C.-H. Yu, Y.-C. Hsu, M.J. Buehler, Deep learning model to predict fracture mechanisms of graphene, *npj 2D Mater. Appl.* 5 (2021) 48.
- [15] S.C. Shen, M. Peña Fernández, G. Tozzi, M.J. Buehler, Deep learning approach to assess damage mechanics of bone tissue, *J. Mech. Behav. Biomed. Mater.* 123 (2021), 104761.
- [16] K. Guo, M.J. Buehler, A semi-supervised approach to architected materials design using graph neural networks, *Extrem. Mech. Lett.* 41 (2020), 101029.
- [17] C.H. Yu, Z. Qin, F. Martin-Martinez, M.J. Buehler, A self-consistent sonification method to translate amino acid sequences into musical compositions and application in protein design using AI, *ACS Nano* 13 (2019) 7471–7482.
- [18] C.-H. Yu, M.J. Buehler, Sonification based de novo protein design using artificial intelligence, structure prediction, and analysis using molecular modeling, *APL Bioeng.* 4 (2020), 016108.
- [19] M. Milazzo, M.J. Buehler, Designing and fabricating materials from fire using sonification and deep learning, *iScience* 24 (2021), 102873.
- [20] W. Xie, et al., Variational Autoencoder Bidirectional Long and Short-Term Memory Neural Network Soft-Sensor Model Based on Batch Training Strategy, *IEEE Trans. Ind. Informatics* 17 (2021) 5325–5334.
- [21] I. Habibie, D. Holden, J. Schwarz, J. Yearsley, T. Komura, A Recurrent Variational Autoencoder for Human Motion Synthesis, *Br. Mach. Vis. Conf.* 28 (2017).
- [22] W. Bao, J. Yue, Y. Rao, A deep learning framework for financial time series using stacked autoencoders and long-short term memory, *PLoS One* 12 (2017), e0180944.
- [23] Aage, N. & Johansen, V. E. A 165 Line Topology Optimization Code. <http://www.topopt.mek.dtu.dk/Apps-and-software/Topology-optimization-codes-written-in-Python> (2013).
- [24] S.K. Kauwe, J. Graser, R. Murdock, T.D. Sparks, Can machine learning find extraordinary materials? *Comput. Mater. Sci.* 174 (2020), 109498.
- [25] E. Andreassen, A. Clausen, M. Schevenels, B.S. Lazarov, O. Sigmund, Efficient topology optimization in MATLAB using 88 lines of code, *Struct. Multidiscip. Optim.* 43 (2010) 1–16.
- [26] O. Sigmund, A 99 line topology optimization code written in Matlab, *Struct. Multidisc Optim.* 21 (2001) 120–127.
- [27] M.P. Bendsøe, Optimization of Structural Topology, Shape, and Material, Springer-Verlag, Berlin Heidelberg, 1995, <https://doi.org/10.1007/978-3-662-03115-5>.

- [28] O. Sigmund, J. Petersson, Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima, *Struct. Optim.* 16 (1998) 68–75.
- [29] D.P. Kingma, Lei Ba, J. Adam, A Method for Stochastic Optimization, in: *Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
- [30] M. Abadi, et al., TensorFlow: A system for large-scale machine learning, in: *Proc. 12th USENIX Symp. Oper. Syst. Des. Implement.*, 2016, pp. 265–283.
- [31] Pascanu, R., Gulcehre, C., Cho, K. & Bengio, Y. How to Construct Deep Recurrent Neural Networks. *arXiv Prepr.* 1–13 (2014).
- [32] Graves, A., Fernández, S. & Schmidhuber, J. Multi-Dimensional Recurrent Neural Networks. *arXiv Prepr.* 1–10 (2013).
- [33] Kalchbrenner, N., Danihelka, I. & Graves, A. Grid Long Short-Term Memory. *arXiv Prepr.* 1–15 (2016).