Structural and Multidisciplinary Optimization (2022) 65:135
https://doi.org/10.1007/s00158-022-03224-x

RESEARCH PAPER

Remixing functionally graded structures: data-driven topology optimization with multiclass shape blending

Yu-Chin Chan[1] · Daicong Da[1] · Liwei Wang[1,2] · Wei Chen[1]

Received: 30 November 2021 / Revised: 22 February 2022 / Accepted: 8 March 2022 / Published online: 12 April 2022

Abstract

To create heterogeneous, multiscale structures with unprecedented functionalities, recent topology optimization approaches design either fully aperiodic systems or functionally graded structures, which compete in terms of design freedom and efficiency. We propose to inherit the advantages of both through a data-driven framework for multiclass functionally graded structures that mixes several families, i.e., classes, of microstructure topologies to create spatially-varying designs with guaranteed feasibility. The key is a new multiclass shape blending scheme that generates smoothly graded microstructures without requiring compatible classes or connectivity and feasibility constraints. Moreover, it transforms the microscale problem into an efficient, low-dimensional one without confining the design to predefined shapes. Compliance and shape matching examples using common truss geometries and diversity-based freeform topologies demonstrate the versatility of our framework, while studies on the effect of the number and diversity of classes illustrate the effectiveness. The generality of the proposed methods supports future extensions beyond the linear applications presented.

Keywords Topology optimization · Functionally graded structure · Multiscale · Multiclass · Shape interpolation · Data-driven design

1 Introduction

Multiscale mechanical structures present exciting functionalities and unprecedented performance ranging from global objectives such as light-weighting, thermal conductivity and energy absorption (Da et al. 2019; Jia et al. 2021; Da et al. 2021) to targeted local behaviors, e.g., shape morphing or pattern reconfiguration for soft robots (Mirzaali et al. 2018; Boley et al. 2019) and active airfoils (Lumpe and Shea 2021). To design such complex structures, multiscale topology optimization (TO) has risen to prominence and flourished. While early research focused on periodic microstructures, two types of *heterogenous* designs, in which neighboring microstructural topologies differ from each other, now surpass them in terms of performance: fully aperiodic systems and functionally graded structures (FGS).

In this work, we aim to bridge the freedom of aperiodic designs with the efficiency and smooth interfaces of FGS. We summarize their pros and cons in this section. Both typically follow a homogenization approach where the material properties of each element in the macroscopic structure are replaced by the effective properties of the microstructure at that location. This greatly expedites multiscale design as the performance can be evaluated at the macroscale, albeit with lowered accuracy since heterogeneous structures break the assumption of infinite periodicity (Andreassen and Andreasen 2014). The two approaches differ in that aperiodic systems can be composed of very different microstructures, whereas the change in the topologies and/or volume fractions of neighboring microstructures in FGS are deliberately designed to be continuous.

Although aperiodic systems allow immense design freedom, they come at the cost of explosive problem sizes. Two

Topical Collection: 14th World Congress of Structural and Multidisciplinary Optimization.

Guest Editors: Y. Noh, E. Acar, J. Carstensen, J. Guest, J. Norato, P. Ramu.

Responsible Editor: Josephine Voigt Carstensen

✉ Wei Chen
weichen@northwestern.edu

[1] Department of Mechanical Engineering, Northwestern University, Evanston 60208, IL, USA

[2] State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China

avenues of data-driven methods have emerged to counter this "curse of dimensionality". One assembles microstructures from pre-computed libraries via combinatorial optimization (Schumacher et al. 2015; Panetta et al. 2015), and the other accelerates gradient-based design by creating deep learning (DL) models from massive datasets for dimension reduction and rapid property predictions (Wang et al. 2020a; Kumar et al. 2020; Xiao et al. 2021).

These approaches are powerful, but require large overhead costs to build the datasets and models, and need careful strategies to select reasonably compatible neighboring microstructures. Even then, they may not achieve connectivity on par with FGS. Combined with the use of the effective properties, low connectivity can result in the manufactured performance deviating greatly from the optimized design. These drawbacks mean that, in their current state, data-driven methods are difficult to scale to large systems with complex physics, e.g., nonlinear mechanics. Compared to FGS, such approaches are less suitable when stress concentrations or expensive property simulations must be avoided.

In contrast, some works have indicated that the continuous interfaces and more gradual topological change in FGS may be able to mitigate the errors from homogenization (Schumacher et al. 2015; Garner et al. 2019; Panesar et al. 2018). Functional grading can be further categorized into three camps: continuously varying volume fraction (Wang et al. 2018; Li et al. 2019; Jansen and Pierard 2020), topology (Kumar et al. 2020; Sanders et al. 2021), or a hybrid of both (Wang et al. 2020c; Luo et al. 2021; Xiao et al. 2021). Recent research is shifting towards the latter two, which have demonstrated that expanding the design space to include multiple topology types can considerably improve the structural performance. Our work belongs to the hybrid one, and we define a *microstructure class* as a family of microstructures that possess the same overall topological concept but vary individually by volume fraction. Hence, the works in the last two categories can be termed *multiclass*.

Within existing multiclass methods, the prevailing strategy is to treat FGS design as a multi-material TO problem by allocating each class to its own region with distinct boundaries. This assumes that the interfaces between classes are perfectly connected. As a result, most approaches pre-define a few mutually compatible classes (Wang et al. 2020c; Luo et al. 2021) or fix their connections (Chu et al. 2019; Zhang et al. 2018), which reduces computational cost and complexity but can yield suboptimal solutions.

Similar methods in the general multiscale TO field accomplish connected heterogeneous designs without the above simplifications by: (1) sharing design variables at interfaces (Liu et al. 2019), (2) adding connectivity constraints (Du et al. 2018; Garner et al. 2019), (3) controlling the change in the properties of intermediate microstructures (Zhou et al. 2019), (4) creating geometric gradations during pre- or post-processing (Sanders et al. 2021; Zhou et al. 2019; Zobaer and Sutradhar 2020), and (5) interpolating random field representations of microstructures (Kumar et al. 2020). Of these, Liu et al. (2019), Garner et al. (2019), and Sanders et al. (2021) concurrently design the macrostructure as well as the distributions of multiple microstructures, and only Luo et al. (2021) also optimized the graded volume fractions. Moreover, many do not scale well with the number of classes.

Merging the advantages of the two heterogeneous approaches, we propose a general TO framework for multiclass FGS that achieves smooth transitions between multiple microstructure classes without additional constraints, even if those classes are not compatible initially. The cornerstone of our approach is a novel multiclass shape blending scheme that generates new microstructures from a small set of predefined basis classes while guaranteeing feasibility. We define a design as *feasible* when the microstructures are not only self-connected (i.e., have no disconnected features) but also well-connected to their neighbors. In addition, we desire the microstructures to meet a minimum feature size, as required in some manufacturing techniques. Our framework departs from existing heterogeneous design methods in several ways:

- We create a continuous and low-dimensional microstructure representation using the parameters of our shape blending scheme as design variables. This effectively transforms the microscale problem into a parametric one. While we do require predefined basis classes, this by no means restricts our design, as we allow the regions for each class and the boundaries between them to be blurred so that the microstructures anywhere in the FGS can be novel, i.e., not found in the initial basis classes.
- Our blending scheme integrates naturally into existing TO methods. In this work, we incorporate it with discrete and gradient-based TO, along with a new penalty that promotes diverse designs. These significantly reduce the cost of concurrent design with any number of microscale classes. Thus, through blending, our framework features design freedom near that of aperiodic methods while inheriting the efficiency of functionally graded design.
- The blending parameters serve as effective inputs for neural networks that predict the effective properties of new microstructures and can be re-used in multiple applications, further accelerating design.
- Certain feasibility metrics (e.g., self-connectedness and minimum feature size) can be built into the blending scheme so that they do not need to be included explicitly in the TO problem. Furthermore, the flexibility of using basis classes permits designers to incorporate expert knowledge and eliminates the frustration of choosing compatible classes. If desired, manual selection can be removed altogether by extracting classes from open-

source databases, e.g., using diversity metrics (Chan et al. 2020).

With compliance and shape morphing examples, we demonstrate the efficiency and inherent ability of our approach in designing multiscale structures with continuous transitions. The benefits of multiclass FGS are verified by utilizing both common truss-type and diverse freeform (topology-optimized) basis classes, and by comparing our results to designs in literature.

2 Methodology

In this section, we introduce three crucial components in our approach: (1) multiclass shape blending and global interpolation schemes (Sect. 2.1), (2) neural networks for property prediction (Sect. 2.2), and (3) concurrent multiclass data-driven topology optimization (TO), which ties all of the methods into one framework (Sect. 2.3).

2.1 Multiclass shape blending and smooth interpolation

2.1.1 Background

Our proposed shape blending scheme is heavily inspired by the computer graphics field, where morphing one geometric model into another has long been studied and utilized in, e.g., animation films and video games (Sanchez 2015; Rohra and Kulkarni 2019). These methods have also supported applications like medical imaging (Carballido-Gamio et al. 2005) and metal-forming simulations (Thomas et al. 2020). In fact, evolving geometries through surface representations is the foundation of the level-set TO method (van Dijk et al. 2013), while combining shapes using distance fields is the bedrock of TO algorithms such as movable morphing components (Zhang et al. 2015).

Most closely related to the interest of this paper are blending techniques that use function representations (FReps) of shapes, which parametrize any geometric model as a series of operations (e.g., unions, differences, and intersections) performed on a set of primitives or basis geometries (Sanchez 2015). It is extremely flexible as it allows the bases to be defined by any representation, e.g., meshes or voxels, and any resolution. In our case, we represent our basis microstructure classes as continuous signed distance fields (SDFs), which are implicit function representations similar to level sets. That is, whether the material is solid or void at any arbitrary point (x, y) within a microstructure, $\mathbf{\Phi}(x, y, t)$, is determined by whether the SDF value is positive or negative, respectively. The isovalue t controls the isocontour of the field and therefore enables us to tune volume fractions. With this powerful representation, we can not only generate an entire family of microstructures over a continuous range of volume fractions, but also combine multiple SDFs to create novel classes.

Blending operations to mix SDFs have been studied for decades, beginning with the simple set-theoretic operation for the union of the FReps of two shapes, $\mathbf{\Phi}_1$ and $\mathbf{\Phi}_2$ (Ricci 1973): $\mathbf{\Phi}_{\text{union}} = \max(\mathbf{\Phi}_1, \mathbf{\Phi}_2)$. Since then, many works have improved shape metamorphosis (Sanchez 2015; Eisenberger et al. 2019; Oring et al. 2021), but require intensive user-interaction and computational costs, and are, therefore, intractable or unsuitable for multiscale design.

2.1.2 Multiclass shape blending scheme

Our proposed multiclass shape blending scheme is a combination of two simple techniques: (1) a weighted sum of basis classes based on cross dissolving, and (2) an activated union with the lower feasible bounds of each basis. The entire scheme is differentiable and efficient, which fits in well with gradient-based optimization algorithms.

Crucial to our scheme are basis microstructure classes, which do not need to be mutually compatible. We denote the SDF representations of arbitrary basis classes as $\mathbf{\Phi}_d^B$ for $d = [1, D]$, where the shapes $\mathbf{\Phi}_d^B$ and total number of classes D are defined by the user. Note that while we use truss-type bases in Figs. 1 and 2 as demonstration, our method is applicable to any set of classes.

Prior to blending, we normalize the D basis classes so that they can be mixed fairly and also find their lower feasible bounds. The following pre-processing steps only need to be performed once per set of bases:

1. Choose a common volume fraction, v^*, and find the representative SDF for each basis, $\mathbf{\Phi}_d^* = \mathbf{\Phi}_d^B + t_d^*$, such that it has v^*. We use the well-known bisection algorithm.
2. Find the SDF of the lower feasible bound of each basis, $\mathbf{\Phi}_d^L = \mathbf{\Phi}_d^B + t_d^L$, using any desired feasibility metric. In this work, we set a minimum feature size of 4 pixels for a 50×50 microstructure.

After this one-time process, we can use $\mathbf{\Phi}_d^*$ and $\mathbf{\Phi}_d^L$ repeatedly for our proposed multiclass blending. The first step of the scheme is based on cross dissolving (Rohra and Kulkarni 2019), which can be simply defined as a linear interpolation between the source and target geometries. This can induce a double exposure effect where traces of both shapes co-exist in the blended result. While that causes unnatural morphing of, e.g., human faces, it organically achieves connected transitions between neighboring microstructures.

We express multiclass interpolation as a weighted sum of the bases:

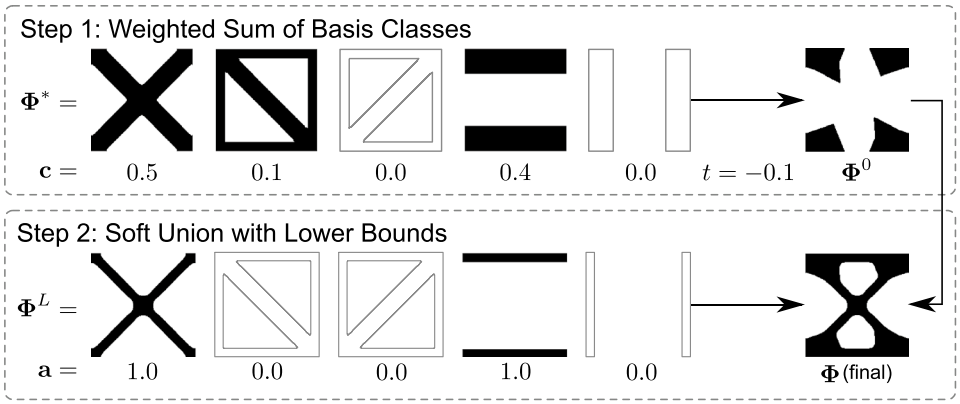


Fig. 1 Illustration of using multiclass shape blending scheme to generate a microstructure from truss basis classes. Outlined shapes represent classes whose weights are zero. Top row: Step 1 (Eq. 1), bottom row: Step 2 (Eq. 2)

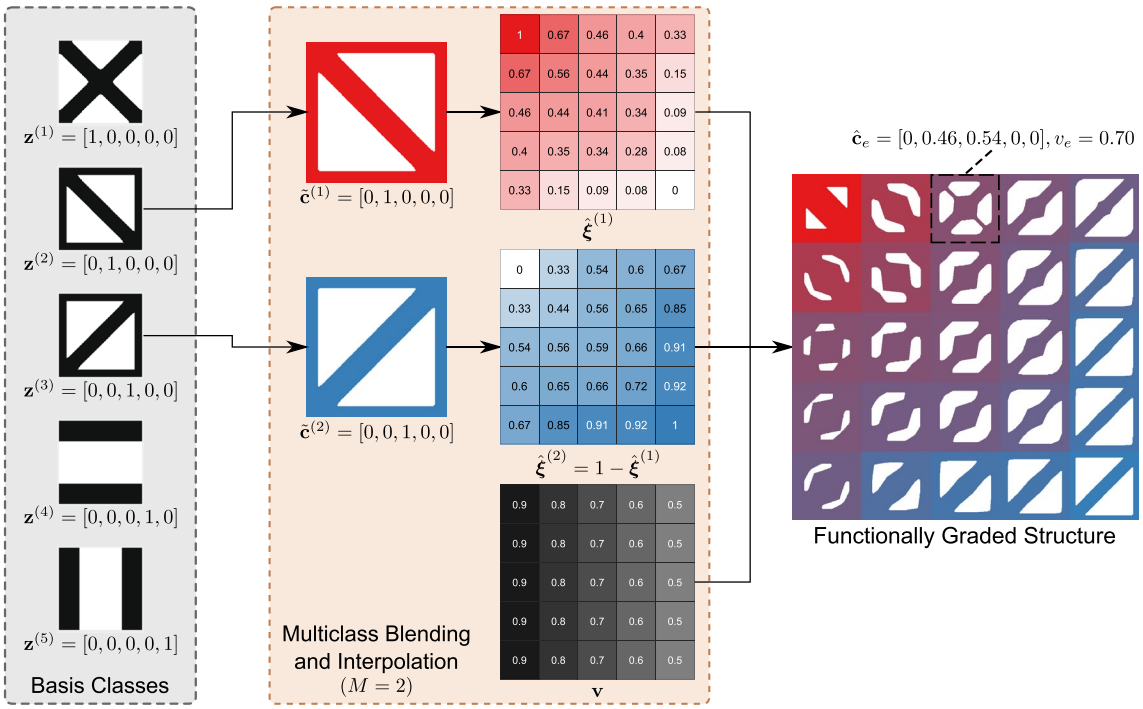


Fig. 2 Demonstration of the integration of multiclass shape blending and global interpolation. The basis classes are defined (left). Examples of $M = 2$ new classes, $\tilde{\mathbf{c}}$, their distributions, $\hat{\boldsymbol{\xi}}$, and volume fractions, $\mathbf{v}$, are given (middle). Each microstructure in the FGS (right) is generated using Eq. 6

$$\mathbf{\Phi}^0 = \sum_d^D c_d \mathbf{\Phi}_d^* + t, \tag{1}$$

where $c_d \in [0, 1]$. Although it effectively creates new classes of microstructures, this interpolation is agnostic to important geometrical features and can lead to broken shapes that have disconnected or thin features (see the top row of Fig. 1).

Therefore, to guarantee that the blended microstructures are sufficiently connected and feasible, we propose an additional step that enforces a lower feasible bound on blending. It also acts as an implicit constraint for simple manufacturing considerations, e.g., minimum feature sizes or volume fractions. This second step is an activated soft-max function, a continuous and differentiable extension of Ricci's set-theoretic union above:

$$\mathbf{\Phi} = \frac{1}{\beta_2} \log \left[\exp\left(\beta_2 \mathbf{\Phi}^0\right) + \sum_d^D a_d \exp\left(\beta_2 \mathbf{\Phi}_d^L\right) \right], \tag{2}$$

where $a_d = H(c_d)$ are the activated weight parameters using the Heaviside function:

$$H(c_d) = \frac{\tanh(\beta_2\eta_2) + \tanh(\beta_2(c_d - \eta_2))}{\tanh(\beta_2\eta_2) + \tanh(\beta_2(1 - \eta_2))}. \tag{3}$$

By setting the threshold $\eta_2 > 0$, the activation guarantees that at least one—but not all, or else low volume fractions would be difficult to attain—of the bases are feasible in each blended result. That is, in this step, only weights that are greater than η_2 are activated to equal one (shaded shapes in the second panel of Fig. 1) while others are suppressed to zero (outlined shapes). We find that setting η_2 to the $75th$-percentile of the weights, $\mathbf{c}$, works quite well in promoting connected transitions.

With the two-step shape blending scheme, the representation of all possible blended microstructures, including those in the original basis classes, can be compactly expressed as the weight parameters. We can, therefore, formulate the microscale design variables as $\mathbf{c}^{(m)}$ for $m \in [1, M]$ desired optimal classes (set by the user), transforming the typically high-dimensional optimization problem into a simple and efficient parametric one that can still generate a wide range of microstructures.

A final note regarding microstructure design is that, in practice, Eq. 1 allows each $c_d^{(m)} \in [0, 1]$ and can lead to $\sum_{d=1}^{D} c_d^{(m)} = 0$, which results in completely solid microstructures and occasional numerical issues. Moreover, it causes redundancy in the design space since taking $\mathbf{\Phi}^0 \geq 0$ to obtain the solid topology cancels out the least common denominator of the weights. There are numerous ways to enforce the sum of $c_d^{(m)}$ to be equal to one. We extend a multi-material interpolation schemes from our previous work (Chan et al. 2019) to normalize the basis class weights as follows:

$$\tilde{\mathbf{c}}^{(m)} = \mathbf{z}^{(1)} + \sum_{j=1}^{D-1} \left[\left(\mathbf{z}^{(j+1)} - \mathbf{z}^{(j)} \right) \prod_{k=1}^{j} c_k^{(m)} \right], \tag{4}$$

where $\mathbf{z}^{(j)}$ are constant one-hot encoded vectors for each basis such that $z_i^{(j)}$ equals 1 for $i = j$ and 0 for all $i \neq j$. Subsequently, c_d in Eqs. 1 and 3 are replaced with $\tilde{c}_d$. While our shape blending and design methods work well without Eq. 4, for increased stability in the optimization process, and the added bonus of reducing the microscale design variables, $\mathbf{c}^{(m)}$, to size $[1 \times D - 1]$, we use it in the remaining discussions.

2.1.3 Integration with multiscale design

In the context of multiscale design, the subscript e is added to denote individual microstructures, which each resides in one macroscopic quadrilateral 4-node finite element. Instead of directly optimizing D weights at each microstructure, we reduce the number of design variables by optimizing M new classes and interpolating them throughout the global structure using the distribution fields $\boldsymbol{\xi}^{(p)}$. An example with $M = 2$ is portrayed by the red and blue classes in the middle panel of Fig. 2. For demonstration, we artificially create the values of the macro- and micro-scale design variables in the middle panel, but during design, these values are optimized concurrently.

Similar to multi-material TO (and Eq. 4), we require that the sum of the distributions at each element equals one. Thus, to obtain each microstructure, e, we can globally interpolate the optimal class weights, $\tilde{\mathbf{c}}^{(m)}$, with:

$$\hat{\mathbf{c}}_e = \tilde{\mathbf{c}}^{(1)} + \sum_{j=1}^{M-1} \left[\left(\tilde{\mathbf{c}}^{(j+1)} - \tilde{\mathbf{c}}^{(j)} \right) \prod_{k=1}^{j} \hat{\xi}_e^{(k)} \right], \tag{5}$$

where $\hat{\boldsymbol{\xi}}$ are the smoothed distribution fields after applying a radial filter (Sigmund 2007) to encourage functional grading.

Therefore, combining Eqs. 1 through 5, our final multiclass shape blending scheme for a microstructure at element e is:

$$\mathbf{\Phi}_e = \frac{1}{\beta_2} \log \Big\{ \exp\left[\beta_2 \Big(\sum_{d}^{D} \hat{c}_{e\cdot d} \mathbf{\Phi}_d^* + t_e \Big)\right] + \sum_{d}^{D} a_{e\cdot d} \exp\left(\beta_2 \mathbf{\Phi}_d^L\right) \Big\}, \tag{6}$$

where $a_{e\cdot d} = H(\hat{c}_{e\cdot d})$ and t_e is found using the bisection algorithm to match a given or optimized volume fraction, v_e. This replaces Eq. 2 during optimization.

2.2 Property prediction with neural networks

The continuous and low-dimensional microstructure representation lends itself well to one of the simplest deep learning (DL) techniques: regression with neural networks. More specifically, we can create feedforward neural networks with three or fewer hidden layers that predict the components of a microstructure's effective stiffness tensor, $\mathbf{C}_e^H$, given the scalar values of the blending weights and volume fraction as inputs. That is, $\mathbf{C}_e^H = NN(\hat{\mathbf{c}}_e, \hat{v}_e)$, where $\hat{\mathbf{c}}_e$ and $\hat{v}_e$ are the interpolated class and filtered volume design variables, respectively.

After each hidden layer, we use a tanh activation function. For training, we use the mean squared error (MSE) loss and the Levenberg-Marquardt optimizer (Moré 1978). To include all possible microstructures, such as those where only a few basis classes have non-zero weights, we use an optimal sliced Latin hypercube method (Ba et al. 2015) to sample the weights c_d, first creating combinatorial "slices", then 20 space-filling samples for each "slice". To cover volume fractions, we also sample 15 microstructures from each resulting SDF (i.e., each set of weights) over $t \in [-1, 1]$. We obtain a total of 22, 575 microstructures for $D = 5$ basis classes. The effective stiffness

tensors of each are calculated using an energy-based homogenization method (Andreassen and Andreasen 2014). We set aside 70%, 15%, 15% of the data for training, validation and testing.

Once the model is trained, we can use backpropagation (Hastie et al. 2009) to analytically derive the gradients of $\mathbf{C}_e^H$ with respect to the design variables, $\hat{\mathbf{c}}_e$ and $\hat{v}_e$. This, together with the rapid predictions that bypass the cost of homogenization, allows the neural networks to significantly boost the efficiency of design.

2.3 Concurrent multiclass data-driven topology optimization

One challenge in creating a concurrent functionally graded design framework that produces realistic results while remaining as general as possible is the different feasible ranges of arbitrary microstructure classes. Consider the five truss basis classes in Sect. 2.1. For a prescribed minimum feature size of 4 pixels, the first basis has a minimum feasible volume fraction of 0.2, while the second has a minimum of 0.4. The question that arises is: when different $v_{\min} > 0$ are possible for each element e, how can we design the distribution of volume fractions while also optimizing a clearly defined macrostructure where some microstructures are allowed to be void (i.e., $v_e = 0$)?

Regarding this, most existing FGS research have either elected to ignore the macroscale design altogether (Li et al. 2019), or adopted a hybridized method that splits the macro- and micro-scale designs into two optimization problems (Wang et al. 2018; Jansen and Pierard 2020; Chu et al. 2019; Zhang et al. 2018). None of these works incorporate multiclass designs where the basis topologies can be drastically different, however.

We propose to overcome this hurdle by merging parametric and non-parametric methods in a framework that utilizes evolutionary TO (BESO (Huang and Xie 2007)) to optimize the discrete global structure, $\mathbf{x}$, and gradient-based TO solved by the method of moving asymptotes (MMA) (Svanberg 1987) to concurrently design the coefficients of M new classes $\mathbf{c}$, their distributions $\boldsymbol{\xi}$, and volume fractions $\mathbf{v}$. The approach is similar to the latter group above, but unlike many, we evolve the designs at both scales in the same iteration. This combination allows arbitrary sets of basis microstructures to be used rather than strict constraints or careful handpicking, and distinguishes our framework in terms of generality and efficiency.

Thus, the general optimization problem is:

$$
\begin{aligned}
\underset{\mathbf{c},\mathbf{v},\boldsymbol{\xi},\mathbf{x}}{\text{minimize}} \quad & f = f_{\text{perf}}(\mathbf{c}, \mathbf{v}, \boldsymbol{\xi}, \mathbf{x}) + k f_{\text{div}}(\mathbf{c}), \\
\text{subject to} \quad & \mathbf{KU} = \mathbf{F}, \\
& g_j \le 0, \\
& 0 \le c_d^{(m)} \le 1, \\
& 0 \le v_{\min} \le v_e \le v_{\max}, \\
& 0 \le \xi_e^{(p)} \le 1, \\
& x_e \in \{x_{\min}, 1\}, \\
& j \in [1, N_{\text{con}}], \quad e \in [1, N_{\text{el}}], \\
& m \in [1, M], \quad d \in [1, D-1], \\
& p \in [1, M-1],
\end{aligned}
\tag{7}
$$

where f_{perf} is an application-dependent measure of design performance, f_{div} is a penalty on low class diversity (Sect. 2.3.1), N_{con} is the number of constraints g_j (if any), N_{el} is the number of macroscopic elements or microstructures, and M and D are the numbers of new (to be optimized) and basis (fixed) classes, respectively. A small number, $x_{\min} = 1\text{e}{-9}$, is used to indicate void microstructures to avoid numerical issues. The minimum volume fraction, $v_{\min}$, is dependent on the chosen set of basis classes, i.e., $\min(volume(\boldsymbol{\Phi}_d^L))$, whereas the upper bound, $v_{\max}$, is 0.95.

We employ the traditional radial averaging filter (Sigmund 2007) on our global-level design variables to avoid mesh dependency, resulting in the smoothed fields $\hat{\mathbf{v}}$, $\hat{\boldsymbol{\xi}}^{(p)}$, and $\hat{\mathbf{x}}$. This additionally enforces the interface between optimal classes to be functionally graded, and that the macrostructure has a minimum feature size of $r_{\min}$.

To update the designs, we use the default algorithms for BESO (Huang and Xie 2007) and MMA (Svanberg 1987). The only difference is that, to connect the two scales, the sensitivity number for BESO are dependent on the predicted effective properties of the microstructures. The derivations for this and all other sensitivities in our problem are shown in Appendix C.

In total, then, our multiclass FGS design framework has $(D-1)M + (M+1)N_{\text{el}}$ variables, where $M \le D \ll N_{\text{el}}$. For the $M = 2$ MBB beam example later (Fig. 4a), our method has 1928 design variables. The multilattice approach by Sanders et al. (2021) uses 3200 variables for the same problem without optimizing the graded volume fractions, while the latent variable multiclass approach of Wang et al. (2020c) has 1920 variables and includes functionally graded volumes; both, however require predefined classes with manually defined connections. It is important to note that although it is possible for other methods to contain less design variables, they make simplifications that we do not.

2.3.1 Penalty to encourage convergence to diverse classes

Our method can enable high design freedom even with a low-dimensional microscale representation since blending allows the basis classes to mix continuously at both scales. Depending on the chosen optimizer, managing such complexity in two-scale design can be a challenge, one that is also encountered by existing multiscale methods. However, we propose that a cost-effective penalty on the objective function can aid the optimizer (in our case, MMA) without resorting to user-defined restrictions on the design space.

We introduce a penalty on low diversity between the M new classes, encouraging the microscale design variables, $\mathbf{c}^{(m)}$, to converge to values away from each other, so that the FGS is more likely to include different basis classes:

$$f_{\text{div}} = -\log\left[\det(L_{ij}(\mathbf{c}^{(i)}, \mathbf{c}^{(j)}))\right], \tag{8}$$

where $L_{ij} = \exp(-0.5\ \|\mathbf{c}^{(i)} - \mathbf{c}^{(j)}\|_2^2)$ and $i, j \in [1, M]$.

This is based on determinantal point processes (DPPs), which measure the diversity of a set of items (e.g., the classes $\mathbf{c}^{(m)}$ here) using a similarity matrix L_{ij}, whose elements are the similarities between i-th and j-th pairs of data. The diversity can then be defined as the determinant of L_{ij}. A larger determinant value indicates that a set contains less similar items, spans a larger volume, and hence has greater diversity. A deeper dive into the benefits of diversity for data-driven multiscale design, can be found in our previous work (Chan et al. 2020).

Intuitively, minimizing f_{div} is equivalent to maximizing the diversity of the new classes. Since the value of f_{div} approaches zero as classes become more diverse, i.e., the values of $c_d^{(i)}$ and $c_d^{(j)}$ grow farther apart, diversity serves as a natural penalty function. It needs only a weight k so that its value, typically within [0, 1] after the first few iterations, can compete with the structural performance, f_{perf}. Moreover, it acts similarly to an L_2 regularizer that smooths the objective function as k increases, which may avoid sensitivity to initializations and help find an optimum faster in some non-convex or highly nonlinear problems. Indeed, we find in our case studies that adding the penalty help our optimizers to find more optimal solutions (Sect. 3.2).

2.3.2 Volume relaxation and adaptive target volume

If any volume constraints are defined in the design problem, we must have a way to obtain the continuous gradients of volume with respect to the parameters of the shape blending scheme. To achieve this, we can approximate the filtered volume fraction of a microstructure, $\hat{v}_e$, by transforming its SDF into a relaxed grayscale field, similar to that of density-based TO, using the Sigmoid function

$$S(\mathbf{\Phi}, \beta_1) = \frac{1}{1 + \exp(-\beta_1 \mathbf{\Phi})}, \tag{9}$$

where β_1 is a fixed parameter to control the strength of relaxation. Thereafter, the approximate volume is

$$\hat{v}_e^a = \frac{1}{n_{\text{el}}} \sum_{u=1}^{n_{\text{el}}} S(\Phi_{e\cdot u}, \beta_1), \tag{10}$$

where n_{el} is the number of elements in the discretized SDF, and the sensitivity of $\partial V_{\text{Global}}/\partial \hat{\mathbf{c}}_e$ is straightforward to calculate (see Appendix C).

In addition, low volumes are often a goal in multiscale design to take advantage of the porosity of the microstructures. By immediately applying a strict volume fraction constraint, however, it is possible to encounter infeasible and broken structures early in the optimization process (Christiansen et al. 2013). To avoid this, and to ensure that our macro- (BESO) and micro-scale (MMA) designs evolve at approximately the same rate, we use an adaptive scheme to lower the target volumes every 10 iterations. The algorithms of this scheme as well as our complete concurrent design framework can be found in Appendix A.

3 Illustrative examples

Through several linear elastic problems, we test the ability of our framework to achieve smooth and feasible functional grading of microstructure morhpologies. Namely, we design two compliance and one shape morphing examples. For each case, we study the effects of two sets of basis classes with different morphology types and initial mutual compatibility, as well as the number of new optimal classes, M.

3.1 Basis classes and neural networks

To illustrate the framework across a range of microstructure morhpologies commonly found in literature, we use two sets of basis classes: one consisting of trusses, and one of topology-optimized freeform shapes. Moreover, to show that human bias can be removed from the design without sacrificing much performance, we compare handpicking the truss basis to automatically selecting the freeform basis using diversity metrics.

3.1.1 Handpicked simple trusses

Truss-type microstructures possess both simple definitions and satisfactory performance (Panetta et al. 2015; Wang et al. 2020c; Luo et al. 2021; Chan et al. 2019). As such, they are fitting basis classes to validate the proposed framework. For our examples, we choose a set that can, when

combined in various ways, cover nearly all of the common truss morphologies in literature. Departing from other methods, we dial up the difficulty by defining the last two bases so that they are broken; to obtain feasible designs, these classes need to be either well-connected to their neighbors or blended with other bases that have self-connectedness. The five classes are shown in Fig. 3a, b.

3.1.2 Shape and property diverse freeform subsets

We also assess the efficacy of our blending and interpolation schemes under even more challenging circumstances by defining a set of freeform basis classes with complex shapes derived from TO (Fig. 3d). They present interesting and highly illustrative case studies as their compatibility with each other is quite low. If used directly in design without our shape blending scheme, the feasibility of the final design would be extremely challenging to guarantee.

We collect these freeform classes from a different perspective, one where a designer has little prior knowledge and wishes to avoid using costly inverse optimization to find the basis microstructures. Thus, the five freeform classes are chosen by leveraging the open-source 2D metamaterials dataset (Wang et al. 2020b) and the automated diverse subset selection method (Chan et al. 2020) from our previous works. The method utilizes the DPPs introduced earlier (Sect. 2.3.1) to maximize the shape and property diversity of a subset of microstructures. By automatically covering a wide range of shapes and properties, we hypothesize that diverse basis classes can provide a high return on investment, attaining competent or even superior performance across multiple applications with less effort during the selection of bases.

We filter out any 50×50 microstructures with minimum feature sizes less than 4 pixels prior to applying our subset selection method. This eliminates some of the most complex shapes that provide little benefit for functional grading (e.g., microstructures with thin features that would limit the range of feasible volume fractions). To convert the selected binary microstructures into continuous SDF representations, we use the fast marching method (scikit-fmm 2021). The shape and property diverse freeform basis classes shown in Fig. 3d, e.

3.1.3 Property prediction models

As discussed in Sect. 2.2, we obtain training datasets of 22, 575 microstructures using sliced Latin hypercube sampling for each set of basis classes. The respective property spaces are depicted in Fig. 3c, f, where samples generated from blending are in gray and those from the original basis classes are denoted as different color crosses.

We observe that the blended microstructures are able to interpolate between—and in some cases, extend slightly beyond—basis classes in order to cover the property space. Although some sparse areas still exist due to the lower feasible bounds that we impose, this shows that blending is a powerful technique to create a large design space even with small sets of basis classes. It is also clear from these figures that the first basis (red) from both sets possess the highest stiffness in diagonal directions and ratios of C^H_{21} to C^H_{11} (equivalent to the effective Poisson's ratio), as opposed to truss bases 4 (purple) and 5 (orange), and freeform basis 2 (blue), which are stiffest in uniaxial directions.

While the freeform classes are orthotropic, some of the truss classes are not. The number of responses, n_{resp}, in the neural networks of each set are adjusted accordingly, i.e., 6 elastic tensor components for trusses and 4 for freeform. Table 1 lists the details of the ML model architectures, where n_{node} indicates the neurons of each hidden layer, and the R^2 and MSE metrics of the trained models.

Since they only need to be built once, the same models are used for all examples throughout the paper. The one-time

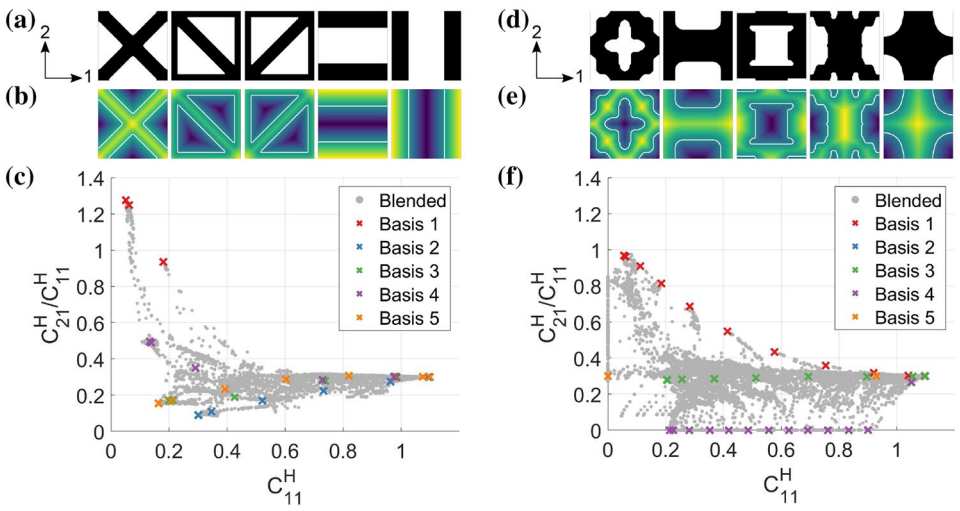


Fig. 3 **a** Truss basis classes represented as **b** SDFs (Φ^*_d), and **c** the property space of 22,575 blended microstructures by sampling $\{\mathbf{c}, v\}$. **d** Freeform basis classes, **e** their SDFs, and **f** property space

Table 1 Neural network architectures and accuracies

	n_{resp}	n_{node}	Train R^2 (MSE)	Val. R^2 (MSE)	Test R^2 (MSE)
Truss	6	{16,16,12}	0.9983 (2.34e–4)	0.9984 (2.80e–4)	0.9983 (2.46e–4)
Freeform	4	{12,12,6}	0.9991 (2.91e–4)	0.9990 (2.90e–4)	0.9991 (3.01e–4)

expense of creating our data and models is reasonable for our 2D problems. However, we note that our results show overlapping properties in the dataset (Fig. 3c, f), high R^2 values above 0.99 and low MSE (Table 1), suggesting that it may not have been necessary to use as many samples as we did. There is great potential to develop adaptive sampling algorithms that better balance accuracy and efficiency, particularly for 3D and complex applications. We leave this for future works.

3.2 Compliance minimization

We begin with compliance minimization examples, the first of which is the classic MBB beam. The boundary conditions are depicted in 4a, and we follow the same set-up as in Xia and Breitkopf (2014) and Wang et al. (2020c) to compare our results with those of existing methods. That is, the MBB beam is discretized into 40×16 microstructures and an ambitious global volume fraction limit is set as $V^*_{\text{Global}} = 0.36$. For the second problem, we pursue a 60×30 bridge structure loaded in three places, as shown in Fig. 4b. Due to symmetry, we can cut its size by half into 30×30. The target global volume there is $V^*_{\text{Global}} = 0.50$.

The two compliance problems can be formulated as:

$$\begin{aligned} \underset{\mathbf{c},\mathbf{v},\boldsymbol{\xi},\mathbf{x}}{\text{minimize}} \quad & f = f_c + kf_{\text{div}}(\mathbf{c}), \\ \text{subject to} \quad & \mathbf{KU} = \mathbf{F}, \\ & g_1 = V_{\text{Global}}/V^*_{\text{Global}} - 1 \le 0, \\ & g_2 = V_{\text{BESO}}/V^*_{\text{BESO}} - 1 \le 0, \end{aligned} \tag{11}$$

where $f_c = \sum_{e=1}^{N_{\text{el}}} \mathbf{u}_e^T \mathbf{k}_e^H(\hat{\mathbf{c}}_e, \hat{\nu}_e)\mathbf{u}_e$ is the compliance with element displacements $\mathbf{u}_e$ and effective stiffness matrices $\mathbf{k}_e^H$, which are predicted via the DL models. The global and macroscale volumes are $V_{\text{Global}} = \sum_{e=1}^{N_{\text{el}}} x_e \hat{\nu}_e / N_{\text{el}}$ and $V_{\text{BESO}} = \sum_{e=1}^{N_{\text{el}}} x_e / N_{\text{el}}$, respectively, and the bounds on the design variables are the same as described in Eq. 7. The sensitivities of this optimization problem are detailed in Appendix C.

For both examples, we initialize the volume constraints as $V^*_{\text{Global},0} = 0.95$, $V^*_{\text{BESO},0} = \sqrt{V^*_{\text{Global}}}$, the volumes $\mathbf{v}$ as 0.95, the class weights so that $\tilde{\mathbf{c}}^{(m)}$ are the same, and all distribution fields, $\boldsymbol{\xi}^{(p)}$, so that the classes are distributed equally. The penalty parameter (Sect. 2.3.1) is set so that $kf_{\text{div}} = 10$ during later iterations. We find that keeping the penalty around this value improves both convergence and design performance.

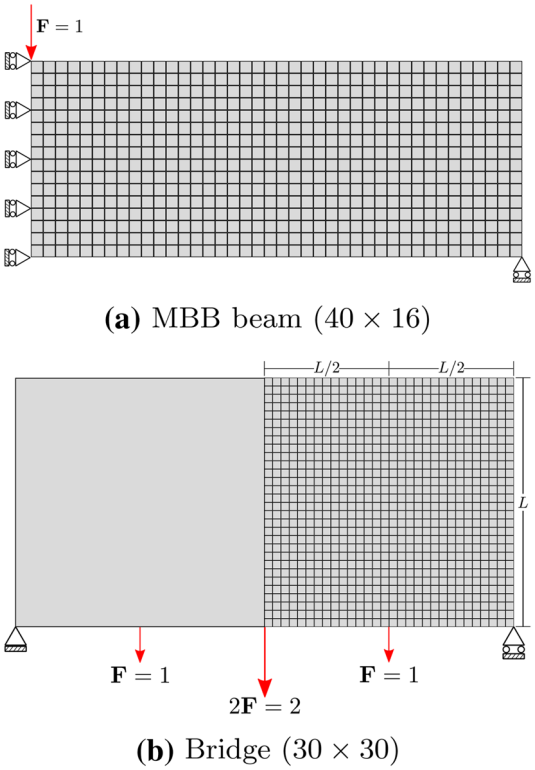


Fig. 4 Problem settings of the compliance minimization examples

The fixed parameters in the shape blending scheme are $\beta_1 = 64$ and $\beta_2 = 32$. As we suggested in Sect. 2.1.2, the threshold η_2 is adapts to the design and is equal to the $75th$-percentile of the class weights. The radii of all filters on the design variables are the same, $r_{\min} = 3.0$, matching Xia and Breitkopf (2014). For BESO, the evolutionary rate is $ER = 0.05$ in both problems. Otherwise, all other parameters are kept at the default values (Huang and Xie 2007; Svanberg 1987). Our convergence criteria are when the change in design or the mean change in the objective over 10 iterations are less than 0.01, or when the number of iterations reaches 200. We also use early-stopping if the target V^*_{Global} has been met but the objective has not improved in 20 iterations.

In the following sections, we present our results in figures with the same layout: The left sides illustrate how the optimized classes are created via blending by breaking them down into the individual basis classes. For ease of interpretation, we show $\tilde{c}_d^{(m)}$ from Eq. 4, which correspond directly to the weights used during blending

(Eq. 6), instead of the design variables $c_d^{(m)}$. The rightmost sides show the optimal multiclass FGS and their final compliance.

For fair comparison between our results and homogenization-based methods in literature, and between the two types of basis morphologies, all compliance values stated in the main paper are calculated using numerical homogenization. For further validation, Tables 5 and 6 in Appendix B also report the compliance obtained from the neural networks and fine mesh analysis.

3.2.1 2-Class results with different basis classes

We first consider the results using $M = 2$ new classes and both sets of basis classes. The optimal designs are shown in Figs. 5 and 6 for the MBB beam with truss and freeform bases, respectively, and in Figs. 7 and 8 for the bridge. The interpretation of the figures is described in the previous section.

From the 2-class compliance results, we can see several benefits of integrating multiclass shape blending into FGS design:

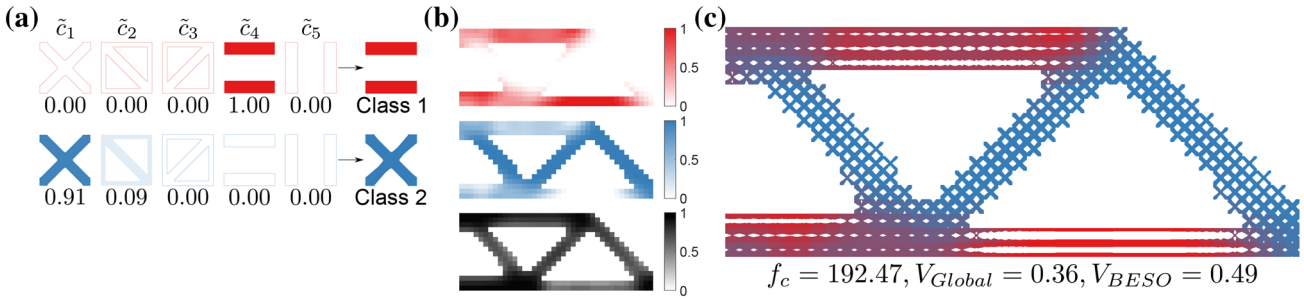


Fig. 5 Truss MBB, 2-class result. **a** Optimal new classes each drawn in a different color. Left of arrows: optimal weights listed under each basis. Lighter colors indicate low weights while outlined shapes represent weights that are zero. Right of arrows: representative topologies of new classes. **b** $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\mathbf{v}}$ from top to bottom, and **c** multiclass FGS

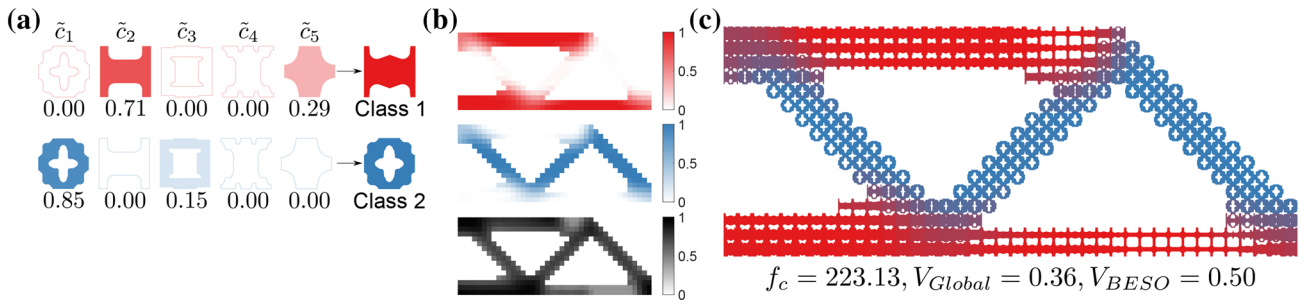


Fig. 6 Freeform MBB, 2-class result: optimal **a** weights and representatives of new classes, **b** $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\mathbf{v}}$ from top to bottom, **c** multiclass FGS

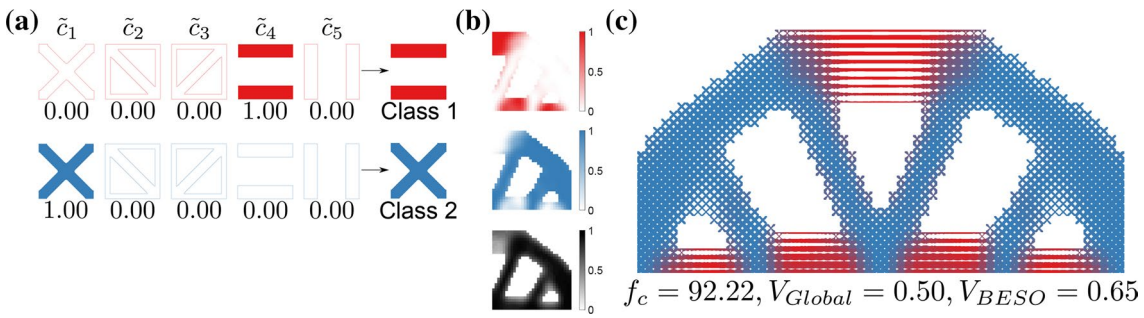


Fig. 7 Truss bridge, 2-class result: optimal **a** weights and representatives of new classes, **b** $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\mathbf{v}}$ from top to bottom, **c** multiclass FGS reflected over the symmetry line

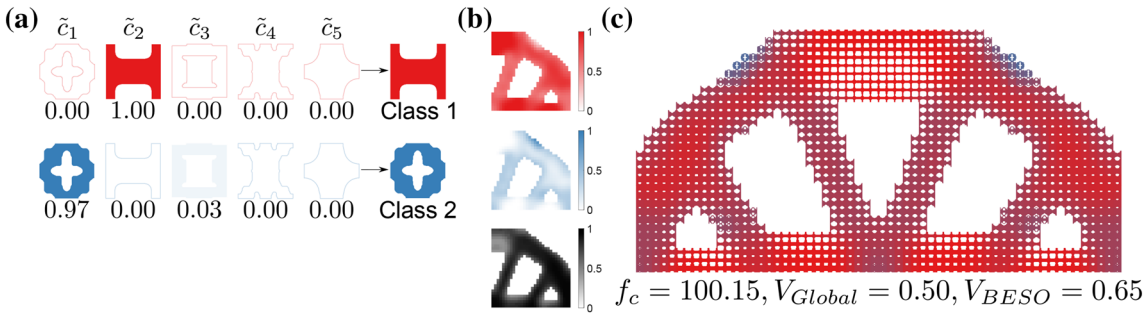


Fig. 8 Freeform bridge, 2-class result: optimal **a** weights and representatives of new classes, **b** $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\mathbf{v}}$ from top to bottom, **c** multiclass FGS reflected over the symmetry line

1. The combination of the blending scheme and the radial filters on the distribution fields creates smooth transitions between classes. Topological functional grading is guaranteed and does not depend on the mutual compatibility of the basis classes. Although connections may not be ideal for our freeform bases, which have low initial connectivity and more complex features, neighboring microstructures change continuously and are at least connected through the imposed lower feasible bounds (Fig. 6).
2. Because of the two-step blending scheme, the microstructures at the interfaces of optimized classes are a union of the classes being mixed there, and the minimum feature sizes of all microstructures match our prescribed lower limit of 4 pixels.
3. The macroscale distributions, $\hat{\boldsymbol{\xi}}^{(p)}$, can be either dominated solely by one class or contain mixtures of multiple classes. For example, the diagonal struts in the truss MBB (Fig. 5a) consist predominantly of the second new class (blue), while the horizontal bars contain both (red and blue), presumably to stiffen the design at those locations. On the contrary, the two new classes in the freeform bridge intermingle throughout nearly the entire structure (Fig. 8).
4. Optimizing $c_d^{(m)}$ can automatically determine if an existing basis class is sufficient to achieve low compliance, or if a novel class needs to be created by fusing several bases. For example, in the freeform MBB result (Fig. 6), the mixture of the second and fifth bases stiffens the microstructures, improving the global compliance of the FGS.
5. Due to BESO, the global macrostructures are clearly defined and change based on the basis classes and optimal microstructures, showing that the hybrid framework works well.

Further observations can be made regarding the framework's ability to adapt to spatially varying stress distributions. The first and fourth truss-type bases, and the first and second for freeform, are the most popular classes by far, agreeing with our observation in Sect. 3.1.3 that these are among the strongest classes in diagonal and uniaxial directions. In both beam and bridge examples, these classes are designed such that the load-bearing features of the blended microstructures intuitively match the load paths.

In particular, our truss-type MBB beam result is akin to those in existing multiscale works with the same design domain. Xia and Breitkopf (2014) performed an exhaustive two-scale TO that optimized every microstructure, resulting in horizontal (uniaxial) and diagonal (anisotropic) features that are oriented with stress directions, and a compliance of $f_c = 190$. Meanwhile, Wang et al. (2020c) proposed a multiclass design with rectangular trusses on the horizontal macro-bars, X's on the diagonal macro-struts, and a compliance of $f_c = 214.02$. Our framework can be thought of as bridging these two methods. This is indeed reflected in our 2-class truss result, which achieves a compliance value between the two existing works, $f_c = 192.47$, and has similar microstructures.

In terms of performance, truss basis classes outshine the freeform basis in both problems. We theorize that our freeform bases do not perform as well for two reasons. (1) They were automatically chosen to maximize diversity, i.e., coverage, in shapes and properties, which undoubtedly can skip microstructures with properties that are more optimal for these specific problems. (2) They contain complex, thin features that force their lower feasible bounds to have high volume fractions ($v_{\min} = 0.28$), which clashes with the low target volume of 0.36 in the MBB problem.

Despite these disadvantages, however, the compliance attained using freeform basis classes is decent across both examples and near those of existing works, confirming the versatility of diverse bases and our design framework. A deeper look into the MBB example also reveals that the multiclass freeform FGS (Fig. 6) still surpasses single-class designs that vary only in volume fraction (Table 2).

Table 2 Single-class MBB beam results using one freeform basis class each

Basis	1	2	3	4	5
f_c	476.26	247.57	384.89	399.29	290.75

3.2.2 Effect of the number of new classes

With the effectiveness of the proposed framework established, we now study whether increasing the number of new classes to $M = 3$ can impact the designs and their performances. The problem definitions remain the same as before. For the MBB beam, the 3-class results are given in Figs. 9 and 10 with the truss and freeform bases, respectively. The bridge results for both sets of bases are combined in Fig. 11, where they are shown in black-and-white and with zoomed-in views of the functionally graded topologies and volume fractions.

The most notable result is the 3-class truss MBB, which achieves a compliance even lower than the fully optimized design of (Xia and Breitkopf 2014) at $f_c = 188.85$. More apparent than in the 2-class result above, it exhibits directional load-bearing features (Fig. 9) such as the left-to-right diagonal microstructures (red) and the near-uniaxial microstructures (blue). In addition, the X-shaped class (green) appears mainly in the middle macro-strut. Overall, our result distinguishes itself from the existing with the mix of isotropic and anisotropic microstructures that are all well-connected. A similar blend of directional and uniaxial classes can be found in our 3-class truss bridge (Fig. 11a–d).

With the exception of the truss-type MBB, however, all 3-class results could not overtake the compliance of their 2-class counterparts. We suspect this is because, for simple compliance problems, additional classes are not necessary to achieve optimal performance. Our hypothesis is supported by the new classes of the $M = 2$ examples: for the most part, they are each monopolized by just one basis class. This suggests that, in most cases, only two basis classes are needed throughout the entire FGS; if more are required, they can be incorporated into a single optimal class by adjusting the values of $\mathbf{c}^{(m)}$ without increasing M, like in the 2-class freeform MBB (Fig. 6). Another reason could be that we force

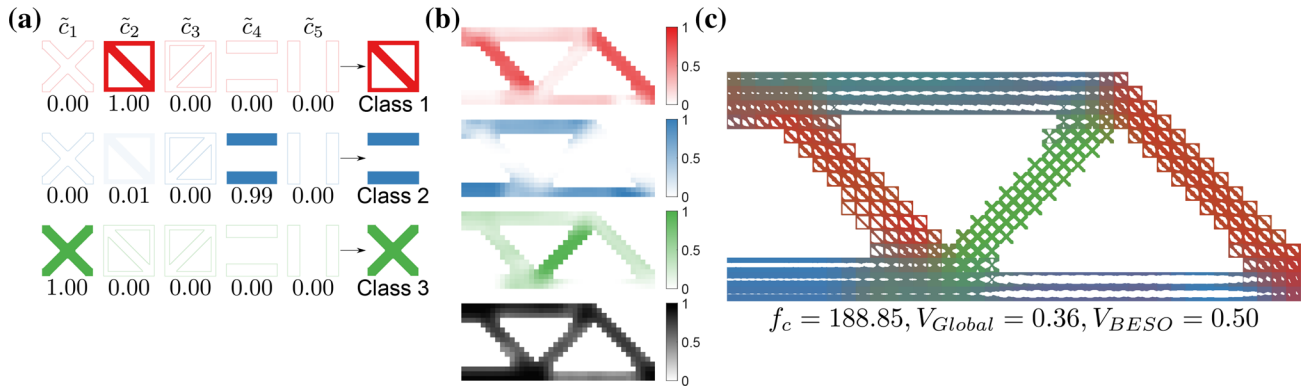


Fig. 9 Truss MBB, 3-class result: optimal **a** weights and representatives of new classes, **b** $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\boldsymbol{\xi}}^{(3)}, \hat{\mathbf{v}}$ from top to bottom, **c** multiclass FGS

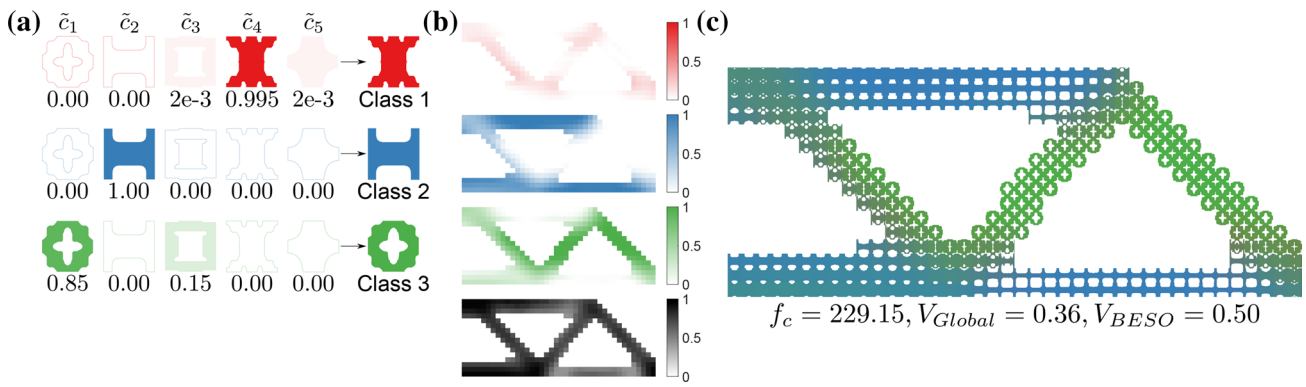


Fig. 10 Freeform MBB, 3-class result: optimal **a** weights and representatives of new classes, **b** $\hat{\boldsymbol{\xi}}^{(1)}, \hat{\boldsymbol{\xi}}^{(2)}, \hat{\boldsymbol{\xi}}^{(3)}, \hat{\mathbf{v}}$ from top to bottom, **c** multiclass FGS

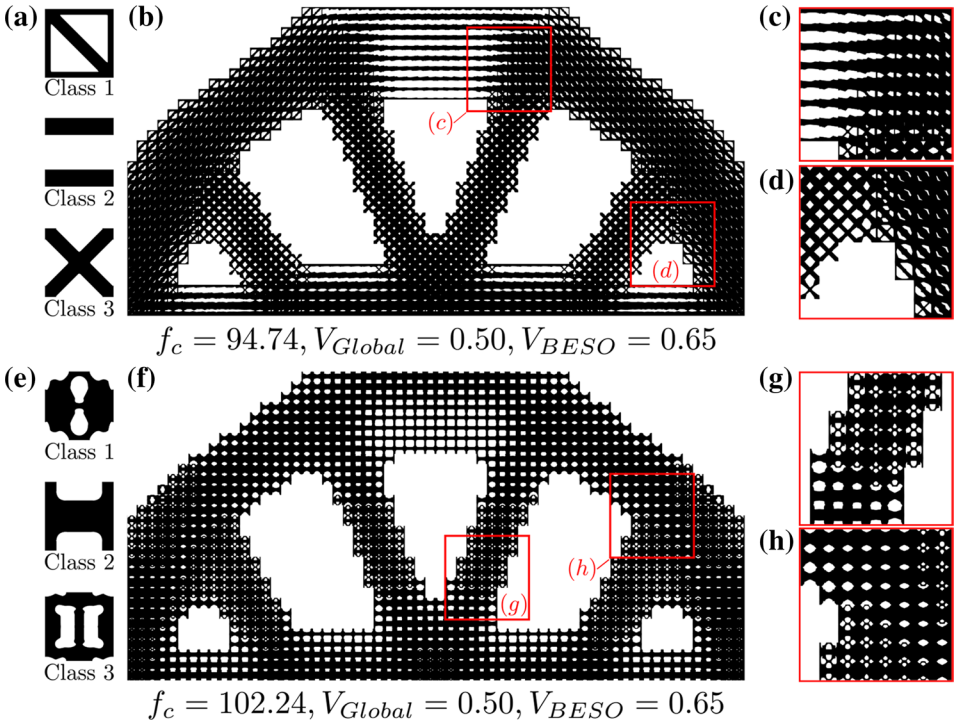


Fig. 11 Results of 3-class bridge in black-and-white. For truss bases: **a** optimal classes, **b** FGS, and **c**, **d** zoom-in views. For freeform bases: **e** optimal classes, **f** FGS, and **g**, **h** zoom-in views

the third class to be different from the others through the penalty on low diversity, which can lead to the addition of a sub-optimal class. This scenario may have occurred in the 3-class freeform MBB, where the first new class (red) is hardly present in the FGS. Nevertheless, the penalty does not significantly worsen the compliance and can, in some cases improve it. An intriguing possibility that bears further investigation is whether a larger variety of microstructures are needed in problems with finer discretization or more complex objectives.

Finally, we compare the computational efficiency of our proposed framework against others, with the caveat that each method was run on different computers. The 40×16 MBB design is reported to require 200 hours in Xia and Breitkopf (2014), and 5 minutes in the data-driven method of Wang et al. (2020c). For our proposed method, the same design using $M = 3$ new classes takes under 12 minutes. However, we note that the majority of this time is consumed by the bisection algorithm (Sect. 2.1.2), which ensures that the microstructures have the optimized volume fractions when converted from SDFs (Eq. 9). Improving this aspect of our blending scheme is next in our future goals.

3.2.3 Effect of the low-diversity penalty

The function that penalizes new classes with low diversity (Eq. 8, Sect. 8) can affect performance, although whether that effect is positive or negative depends on the problem or basis classes. In this section, we show concrete examples why the penalty is still recommended by running the same compliance problems without the penalty, i.e., by setting $k = 0$. The results are listed in Tables 3 and 4. Immediately, we can see that although the truss basis classes can still achieve satisfactory compliance values lower than the existing baselines (Xia and Breitkopf 2014; Wang et al. 2020c), none of these can beat our results above.

In the tables, we write the highest weight values of each new class, $\tilde{\mathbf{c}}_m$, in bold. From this, we observe that each result is often overshadowed by one basis class (see the bold values

Table 3 MBB beam results without penalty on low class diversity ($k = 0$)

Basis	M	$\tilde{\mathbf{c}}_m$	f_c	V_{Global}	V_{BESO}
Truss	2	$\tilde{\mathbf{c}}_1 = [0.00, \mathbf{0.39}, 0.00, 0.27, 0.33]$	205.77	0.36	0.50
		$\tilde{\mathbf{c}}_2 = [0.00, 0.05, 0.11, \mathbf{0.84}, 0.00]$			
	3	$\tilde{\mathbf{c}}_1 = [0.00, \mathbf{0.36}, 0.22, 0.12, 0.30]$	212.83	0.36	0.54
		$\tilde{\mathbf{c}}_2 = [0.12, 0.12, 0.09, \mathbf{0.66}, 0.01]$			
		$\tilde{\mathbf{c}}_3 = [0.23, 0.08, 0.16, \mathbf{0.53}, 0.01]$			
Free-form	2	$\tilde{\mathbf{c}}_1 = [0.00, 0.09, 0.00, 0.00, \mathbf{0.91}]$	252.02	0.36	0.51
		$\tilde{\mathbf{c}}_2 = [0.04, 0.14, 0.39, 0.01, \mathbf{0.43}]$			
	3	$\tilde{\mathbf{c}}_1 = [0.01, 0.00, \mathbf{0.53}, 0.00, 0.46]$	248.84	0.36	0.50
		$\tilde{\mathbf{c}}_2 = [0.00, 0.45, 0.00, 0.00, \mathbf{0.55}]$			
		$\tilde{\mathbf{c}}_3 = [0.00, 0.00, 0.00, 0.00, \mathbf{1.00}]$			

The dominant weights of each new class, $\tilde{\mathbf{c}}_m$, are in bold

Table 4 Bridge results without penalty on low diversity ($k = 0$)

Basis	M	$\tilde{\mathbf{c}}_m$	f_c	V_{Global}	V_{BESO}
Truss	2	$\tilde{\mathbf{c}}_1 = [0.00, 0.22, 0.00, \mathbf{0.73}, 0.05]$	96.59	0.50	0.65
		$\tilde{\mathbf{c}}_2 = [0.00, 0.41, 0.00, 0.09, \mathbf{0.51}]$			
	3	$\tilde{\mathbf{c}}_1 = [0.00, 0.18, 0.00, \mathbf{0.82}, 0.00]$	96.34	0.50	0.65
		$\tilde{\mathbf{c}}_2 = [0.00, 0.28, 0.00, 0.21, \mathbf{0.51}]$			
		$\tilde{\mathbf{c}}_3 = [0.00, 0.45, 0.00, 0.00, \mathbf{0.54}]$			
Free-form	2	$\tilde{\mathbf{c}}_1 = [0.00, 0.00, 0.13, \mathbf{0.87}, 0.00]$	105.90	0.50	0.65
		$\tilde{\mathbf{c}}_2 = [0.03, 0.00, 0.06, \mathbf{0.91}, 0.00]$			
	3	$\tilde{\mathbf{c}}_1 = [0.00, 0.00, 0.10, \mathbf{0.90}, 0.00]$	105.46	0.50	0.65
		$\tilde{\mathbf{c}}_2 = [0.00, 0.00, 0.09, \mathbf{0.91}, 0.00]$			
		$\tilde{\mathbf{c}}_3 = [0.02, 0.00, 0.02, \mathbf{0.96}, 0.00]$			

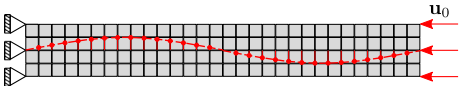
The dominant weights of each new class, $\tilde{\mathbf{c}}_m$, are in bold

in the same column). For the 3-class designs in particular, the second and third new classes are always dominated by the same basis, confirming our earlier suspicion that $M = 2$ is enough to produce optimal results. We also note that there are numerous low values of $\tilde{\mathbf{c}}_m$, signifying that multiple basis classes are being blended into the FGS without improving the design performance. These results additionally imply that greater diversity amongst the microstructure classes improves performance. A more meticulous study on the impact of diversity on the generality and performance of design methods is an intriguing path for future works.

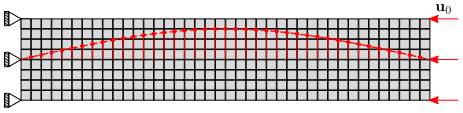
Furthermore, it takes significantly longer for the class design variables to converge without penalization. In the 2-class truss bridge example without the penalty, they often fluctuate and need more than 100 iterations to start converging, whereas they are already converged in under 30 iterations with penalization. These studies validate the benefits that our proposed low-diversity penalty function supply to MMA, helping it to stabilize, escape local minima, and find more optimal solutions. In Appendix B, we provide the convergence plots of the $M = 3$ MBB beam examples, along with additional discussion.

3.3 Shape matching

Heterogeneous structures show great potential for applications where a specific deformation pattern is desired upon actuation, e.g., in form-fitting wearables and soft robots (Mirzaali et al. 2018; Lumpe and Shea 2021; Boley et al. 2019). Motivated by these applications, we optimize two shape matching structures: (1) the target sine-wave



(a) Target sine wave (30×4)



(b) Target bump profile (40×8)

Fig. 12 Problem settings of the shape matching examples

deformation profile shown in Fig. 12a, and (2) the bump profile in Fig. 12b. The first is a cantilever discretized into 30×4 microstructures, fixed at its left side and loaded with displacement boundary conditions on the right—the same example we tested in (Chan et al. 2020). The second is similar, but discretized into 40×8. Like the compliance examples, we will also use the truss and freeform basis classes with $M = \{2, 3\}$.

In past works (Chan et al. 2020; Wang et al. 2020a), we found that these target displacement problems are similar to compliant mechanism design and most effectively solved via a two-stage "top-down" approach that first utilizes inverse TO to find the target effective properties for each microstructure. Departing from previous works, we use the proposed multiclass blending framework in the second stage to optimize the new classes, $\mathbf{c}^{(m)}$, their distributions, $\boldsymbol{\xi}^{(p)}$, and the volume fractions, $\mathbf{v}$, until the target properties are matched.

To find the target properties, i.e., the effective stiffness matrices $\mathbf{C}_t^H$, that achieve a desired displacement profile, the first stage follows the method in Wang et al. (2020b) with following problem:

$$\begin{aligned} \underset{\mathbf{C}_t^H}{\text{minimize}} \quad & \frac{1}{n}\|\mathbf{u} - \mathbf{u}_t\|_2^2, \\ \text{subject to} \quad & \mathbf{KU} = \mathbf{F}, \\ & -\phi(\mathbf{C}_t^H) \le 0, \end{aligned} \tag{12}$$

where $\mathbf{u}$ is the displacement vector of n nodes located on the horizontal centerline of the structure, $\mathbf{u}_t$ is the vector of target displacements of the same nodes, $\mathbf{K}$ is the global stiffness matrix, and $\mathbf{U}$ and $\mathbf{F}$ are global displacement and loading vectors, respectively. This inverse problem uses the stiffness matrices of each microstructure as design variables. To ensure that these are within the bounds attainable by

shape blending, they are constrained by the signed L_2 distance field ϕ of the properties of the training data for the neural networks (Sect. 2.2 and Fig. 3c, f).

After this, the multiclass FGS is optimized to meet the effective property targets by leveraging our proposed blending scheme. Since we do not aim for a target volume here, there is no global volume constraint and the macrostructure defined by $\mathbf{x}$ remains fixed. The second stage is thus:

$$\begin{aligned} \underset{\mathbf{c},\mathbf{v},\boldsymbol{\xi}}{\text{minimize}} \quad & \frac{1}{N_{\text{el}}} \|\mathbf{C}^H(\mathbf{c},\mathbf{v},\boldsymbol{\xi}) - \mathbf{C}_t^H\|_2^2 + kf_{\text{div}}(\mathbf{c}), \\ \text{subject to} \quad & \mathbf{KU} = \mathbf{F}, \end{aligned} \tag{13}$$

where the bounds on the design variables are the same as in the previous examples. Due to the omission of $\mathbf{x}$, it can be solved with just MMA. The penalization parameter is set such that $kf_{\text{div}} = 4$ for the 2-class study and 9 for 3-class, respectively. The filter radius is $r_{\text{min}} = 1.2$ and 2.5 for the sine and bump problems. The volume fractions are initialized at $\mathbf{v} = 0.5$, and all other parameters are the same as in the compliance examples.

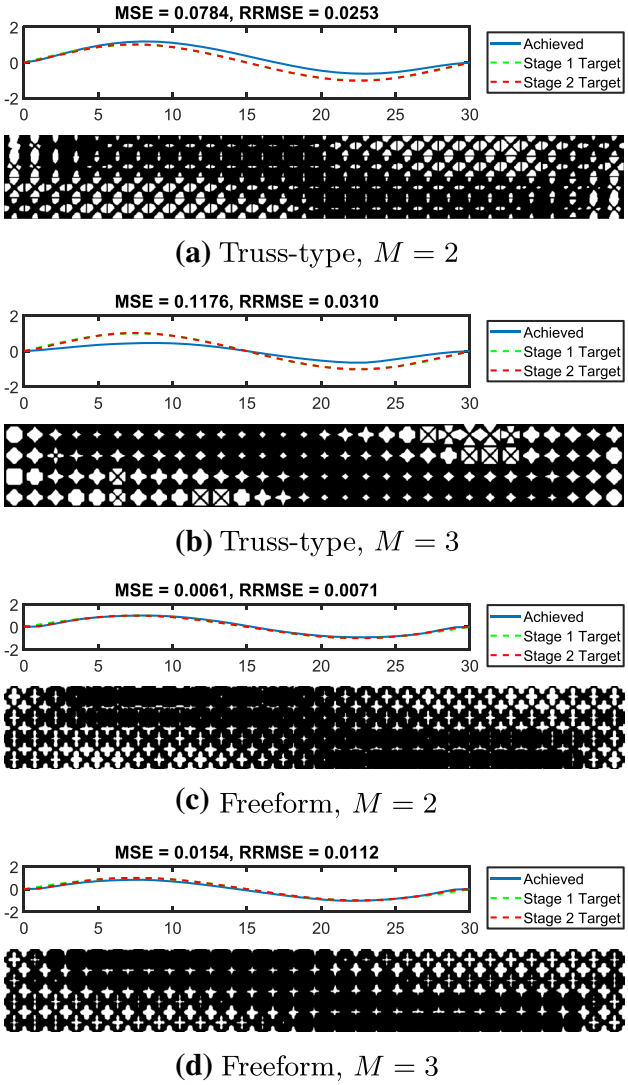


Fig. 13 Results of the target sine wave problem

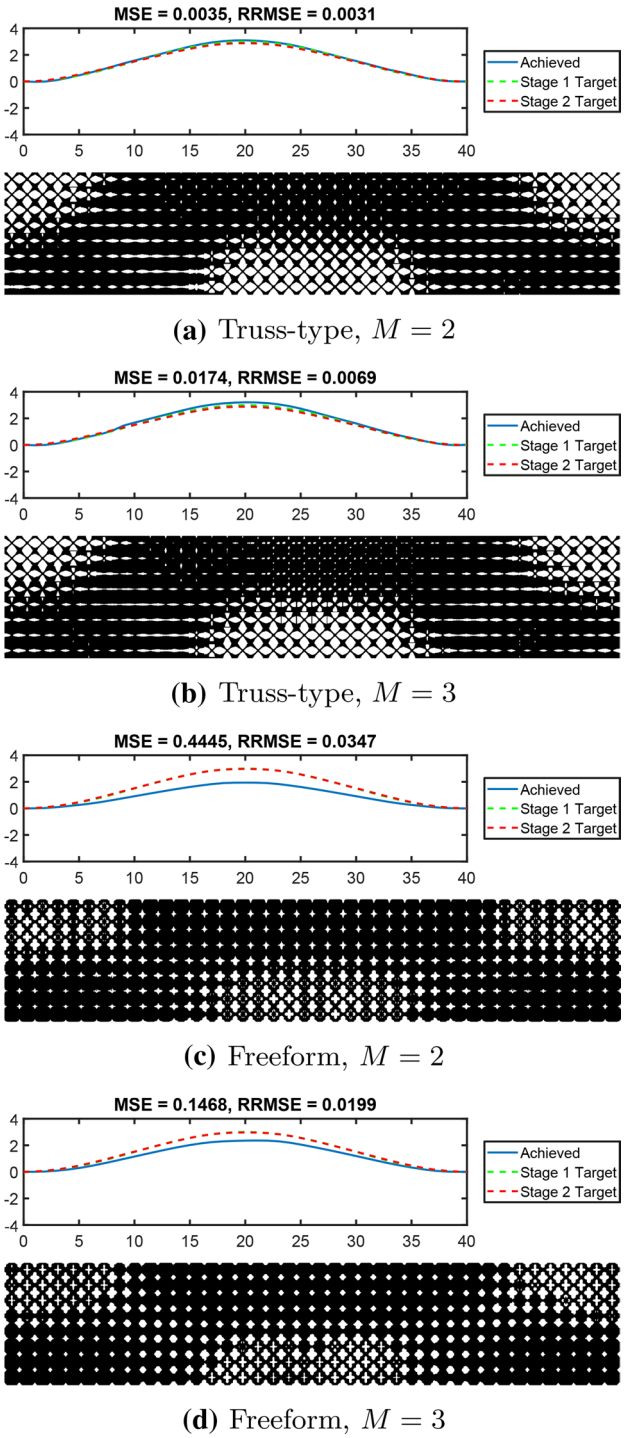


Fig. 14 Results of the target bump shape example

The final multiclass FGS and their achieved displacement profiles (solid blue) are gathered in Fig. 13 for the sine wave and Fig. 14 for the bump problem. In the plots of the displacements, we also show the initial target profile, $\mathbf{u}_t$, used in stage one in dashed green lines, along with profile realized by the optimized properties (dashed red), which serves as an indirect target profile in the second stage.

We performed the same sine wave study in Chan et al. (2020), but with a combinatorial method for aperiodic designs, i.e., without functional grading. There, the lowest MSE that we achieved was 0.1146, which most of our proposed multiclass FGS surpass. Interestingly, the freeform basis classes perform considerably better than the truss-types. The freeform designs are composed mostly of the first basis, which we noted in Sect. 3.1.3 has one of the greatest effective Poisson's ratios, as well as the fifth freeform basis, which has both high stiffness and medium Poisson's ratio.

Conversely, the truss-type FGS match the target bump profile more closely than the freeform ones by utilizing the fourth (horizontal) truss basis. By inspecting the target properties for each problem, we find that the sine wave requires middling values of both Poisson's ratio and stiffness throughout the FGS, which the freeform classes provide more easily, whereas the bump profile needs distinct regions of either large x-directional stiffness or high Poisson's ratio, which the first and fourth truss classes meet exceptionally well (Fig. 3c, f). This observation portends a possible extension of our work where the most efficient basis classes can be chosen to match the initial distribution of principal macroscale stresses for specific problems, similar to Xu and Cheng (2018).

Another intriguing note is that, by blending the last two truss basis classes, we can form square microstructures that are not found in the original set (Fig. 13b). Moreover, combining those two with the 'X' basis creates microstructures with star-shaped voids that are not strictly trusses. These are direct results of the weighted sum of SDFs in our proposed blending scheme (Eq. 1), which can non-intuitively morph the basis classes to achieve optimal performance.

4 Conclusions

We proposed in this work a novel multiclass shape blending scheme that provides a low-dimensional representation of microstructures for both design and DL, and a data-driven multiscale design method that utilizes a hybrid of TO algorithms along with a new penalty on low diversity designs. By integrating these, we created a multiclass FGS design framework that encapsulates the freedom of fully aperiodic structures while featuring efficiency greater than that of typical multiscale methods. The key is the ability of shape blending to blur the lines between classes, creating graded designs with novel microstructures beyond the initial basis classes. Even with classes that have complex features or are not mutually compatible, continuous transitions between neighboring microstructures are guaranteed.

Furthermore, feasibility constraints are incorporated into the scheme to ensure that they are naturally met. In this work, we use a simple measure—minimum feature size. However, defining the lower feasible bounds of each basis class outside of the optimization process means there is potential for future works to incorporate other feasibility, or even quality, metrics, such as those without cheap or tractable gradients.

We demonstrated these advantages through compliance and shape matching examples, in which blending empowered our FGS to surpass designs in literature. Our results revealed that truss-type classes consistently achieve low compliance, and that diverse freeform classes reach satisfactory performance across multiple applications despite being automatically chosen without considering their compatibility. We also discovered that more is not always better when it comes to classes. By encouraging the design to converge to a smaller number of diverse classes, as few as two can be blended to obtain optimal designs. This outcome merits deeper exploration in the future on how diversity metrics can benefit structural design.

Our framework is general in that it is not tied to the specific DL and TO methods shown in this work. It is also not limited to our 2D classes, since multiclass blending is independent of the topology, representation, dimension and resolution of the basis shapes. This modularity is an especially welcome feature as more advanced prediction models and TO algorithms emerge to solve complex multiphysics and nonlinear mechanics problems, including 3D ones. Beyond the examples presented, our framework can be extended to sought-after functionalities like thermoelasticity, fracture resistance and energy absorption, and adapted to applications such as customized user products and architectural design. We believe these are all exciting avenues for future works enabled by multiclass shape blending.

Appendix A: Algorithms

Algorithm 1 Adaptive scheme to decrease volume fraction limits during concurrent multiscale design. i denotes the number of the current iteraton.

Require: $V^*_{Global,i-1}, V^*_{BESO,i-1}, V_{Global}, V_{BESO}$;
1: **if** $(i \mod 10) = 0$ **and** $V^*_{Global,i-1} > V^*_{Global}$ **then**
2: $V^*_{Global,i} \leftarrow \min(V_{Global}, V^*_{Global,i-1}) - 0.025$;
3: **end if**
4: **if** $(i \mod 10) = 0$ **and** $V_{BESO} \leq V^*_{BESO,i-1}$ **then**
5: $V^*_{BESO,i} \leftarrow V^*_{BESO,i-1} - 0.005$;
6: **end if**
7: **return** updated volume constraints.

Algorithm 2 Concurrent design framework for multiclass functionally graded structures. If there are no volume constraints, ignore Line 10. If the macrostructure is fixed, ignore Line 12.

1: **Initialize:**
design variables $\mathbf{c}, \mathbf{v}, \boldsymbol{\xi}, \mathbf{x}$;
volume constraints
$V^*_{Global,0}, V^*_{BESO,0}$;
weight on low-diversity penalty k;
2: **while** change in design $> tol$ **do**
3: $i \leftarrow i + 1$; ▷ iteration counter
4: **for** each macro-element e **do**
5: obtain $\hat{\mathbf{c}}_e$ (Eqs. 4, 5);
6: find $\boldsymbol{\Phi}_e$ and t_e so that unit cell has volume $\hat{v}_e$ (Eq. 6);
7: approximate $\hat{v}^a_e$ (Eqs. 9, 10);
8: predict effective stiffness $\mathbf{C}^H_e = NN(\bar{\mathbf{c}}_e, \hat{v}_e)$ and obtain $\mathbf{k}_e$;
9: **end for**
10: update volume fraction constraint limits (Algorithm 1);
11: compute objective, constraints and sensitivities (Appendix C);
12: update macroscale design $\mathbf{x}$ with BESO;
13: update other variables $\mathbf{c}, \mathbf{v}, \boldsymbol{\xi}$ with MMA;
14: **end while**
15: **return** optimal multiclass functionally graded design.

Appendix B: Additional results

Further examples of blending

To help visualize blending between pairs of basis classes, we illustrate several examples in Fig. 15. In particular, we select pairs that have not appeared in our design results. No matter how simple, complex or incompatible the basis classes, our multiclass blending scheme is able to provide feasible, i.e., well-connected, intermediate microstructures. This is the benefit of (1) the interpolation between the SDFs of basis classes, and (2) the imposition of the lower feasible bounds of each basis through Eqs. 1 and 2.

Compliance validation

In Sect. 3.2, we presented the compliance as calculated by numerical homogenization to directly compare our results to those in literature. In Tables 5 and 6, we additionally report the compliance of our designs performed using: (1) neural networks (same as during design), (2) numerical homogenization (as shown in the main paper), and (3) fine mesh analysis using the multigrid method (Amir et al. 2013) (for validation).

For MBB in particular, the homogenization-based compliance agree reasonably well with the fine mesh analysis. In general, the numerical homogenization-based results, $f_{c,\text{Hom}}$, are slightly closer to the fine analysis values, $f_{c,\text{Fine}}$, than the ones using the neural networks, $f_{c,NN}$, which is expected. However, the differences between $f_{c,NN}$ and $f_{c,\text{Hom}}$ are not large, validating the use of our predictive models to accelerate the design process.

Comparison of convergence with and without penalty

To demonstrate the effect of the low-diversity penalty (Sect. 2.3.1) on convergence, we compare the 3-class MBB examples using both the freeform and truss basis classes (Sect. 3.2.2). This problem had the most difficulty converging to low compliance values. The histories of the class design variables (which are the only variables used to compute the penalty) are plotted in Fig. 16.

The penalty on low class diversity has a significant impact on the convergence behavior of the class design variables. They confirm our statement in Sect. 3.2.3 that applying the penalty can help the weights of the class design variables to converge more quickly, as well as alleviate possible local minima (e.g., compare Figs. 16a, b).

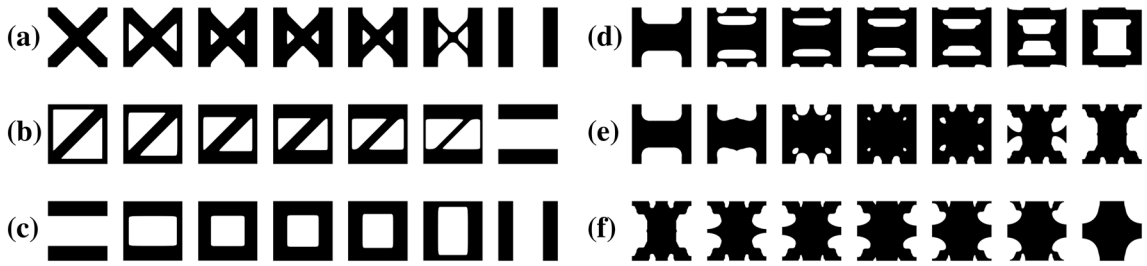


Fig. 15 Examples of blending pairs of basis classes. The first and last microstructures are from basis classes, and those in between are produced by linearly interpolating the weights in the blending scheme. Between truss classes (left column): **a** 1 and 5, **b** 2 and 4, and **c** 4 and 5. Between freeform classes (right column): **d** 2 and 3, **e** 2 and 4, and **f** 4 and 5

Appendix C: Sensitivity analysis

The derivations of the sensitivity analysis for gradient-based topology optimization (TO) are detailed in this section. The nomenclature are the same as in the main text. All derivations were verified using the finite difference method.

For the global multiclass shape blending scheme (Eq. 6), which is used to obtain the microscale topology at e, the gradient is

$$\frac{\partial \boldsymbol{\Phi}_e}{\partial \hat{c}_{e\cdot d}} = \frac{1}{\beta_2 \boldsymbol{\Phi}_d^I}\left[\beta_2 \exp\left(\beta_2 \boldsymbol{\Phi}_e^0\right) \odot \boldsymbol{\Phi}_d^* + \frac{\partial H(\hat{c}_{e\cdot d}, \beta_2, \eta_2)}{\partial \hat{c}_{e\cdot d}} \exp\left(\beta_2 \boldsymbol{\Phi}_d^L\right)\right], \tag{C1}$$

where

$$\boldsymbol{\Phi}_e^I = \exp\left(\beta_2 \boldsymbol{\Phi}_e^0\right) + \sum_d^D a_d \exp\left(\beta_2 \boldsymbol{\Phi}_d^L\right), \tag{C2}$$

$$\boldsymbol{\Phi}_e^0 = \sum_d^D \hat{c}_{e\cdot d} \boldsymbol{\Phi}_d^* + t_e, \tag{C3}$$

$$\frac{\partial H(\hat{c}_{e\cdot d})}{\partial \hat{c}_{e\cdot d}} = \frac{\beta_2\left[1 - \tanh^2(\beta_2(\hat{c}_{e\cdot d} - \eta_2))\right]}{\tanh(\beta_2 \eta_2) + \tanh(\beta_2(1-\eta_2))}, \tag{C4}$$

and $\odot$ indicates element-wise multiplication.

The sensitivities for the the class interpolation schemes (Eqs. 4 and 5) with respect to the class design variables is as follows:

$$\frac{\partial \hat{c}_{e\cdot d}}{\partial c_d^{(m)}} = \left\{ \frac{\partial \tilde{c}_d^{(1)}}{\partial \tilde{c}_d^{(m)}} + \sum_{j=1}^{M-1}\left[\frac{\partial\left(\tilde{c}_d^{(j+1)} - \tilde{c}_d^{(j)}\right)}{\partial \tilde{c}_d^{(m)}} \prod_{k=1}^{j} \hat{\xi}_e^{(k)}\right]\right\} \frac{\partial \tilde{c}_d^{(m)}}{\partial c_d^{(m)}}, \tag{C5}$$

where

$$\frac{\partial \tilde{c}_d^{(m)}}{\partial c_d^{(m)}} = \sum_{j=d}^{D-1}\left[\left(z_d^{(j+1)} - z_d^{(j)}\right) \prod_{k=1, k\neq d}^{j} c_k^{(m)}\right]. \tag{C6}$$

With respect to the macroscale distribution fields, it is:

$$\frac{\partial \hat{c}_{e\cdot d}}{\partial \xi_e^{(p)}} = \sum_{j=p}^{M-1}\left[\left(\tilde{c}_d^{(j+1)} - \tilde{c}_d^{(j)}\right) \prod_{k=1, k\neq p}^{j} \hat{\xi}_e^{(k)}\right] \frac{\hat{\xi}_e^{(p)}}{\xi_e^{(p)}}. \tag{C7}$$

Table 5 Comparison of the compliance of our MBB results in Sect. 3.2.1 calculated by: neural network models $f_{c,NN}$, numerical homogenization $f_{c,\mathrm{Hom}}$, fine mesh analysis $f_{c,\mathrm{Fine}}$

Basis	M	$f_{c,NN}$	$f_{c,\mathrm{Hom}}$	$f_{c,\mathrm{Fine}}$
Truss	2	187.41	192.47	216.50
		(−13.44%)	(−11.10%)	
	3	199.74	188.85	209.28
		(−4.56%)	(−9.76%)	
Freeform	2	210.52	223.13	265.28
		(−20.64%)	(−15.89%)	
	3	212.68	229.15	274.60
		(−22.55%)	(−16.55%)	

For the homogenization-based values, we also report the percent error from $f_{c,\mathrm{Fine}}$ in parentheses

Table 6 Compliance of our bridge results in Sect. 3.2.2 calculated three different ways, including the percent error from $f_{c,\mathrm{Fine}}$ in parentheses

Basis	M	$f_{c,NN}$	$f_{c,\mathrm{Hom}}$	$f_{c,\mathrm{Fine}}$
Truss	2	90.00	92.22	138.29
		(−34.92%)	(−33.31%)	
	3	92.59	94.74	135.57
		(−31.70%)	(−30.12%)	
Freeform	2	95.65	100.15	144.01
		(−33.58%)	(−30.46%)	
	3	93.12	102.24	146.66
		(−36.51%)	(−30.29%)	

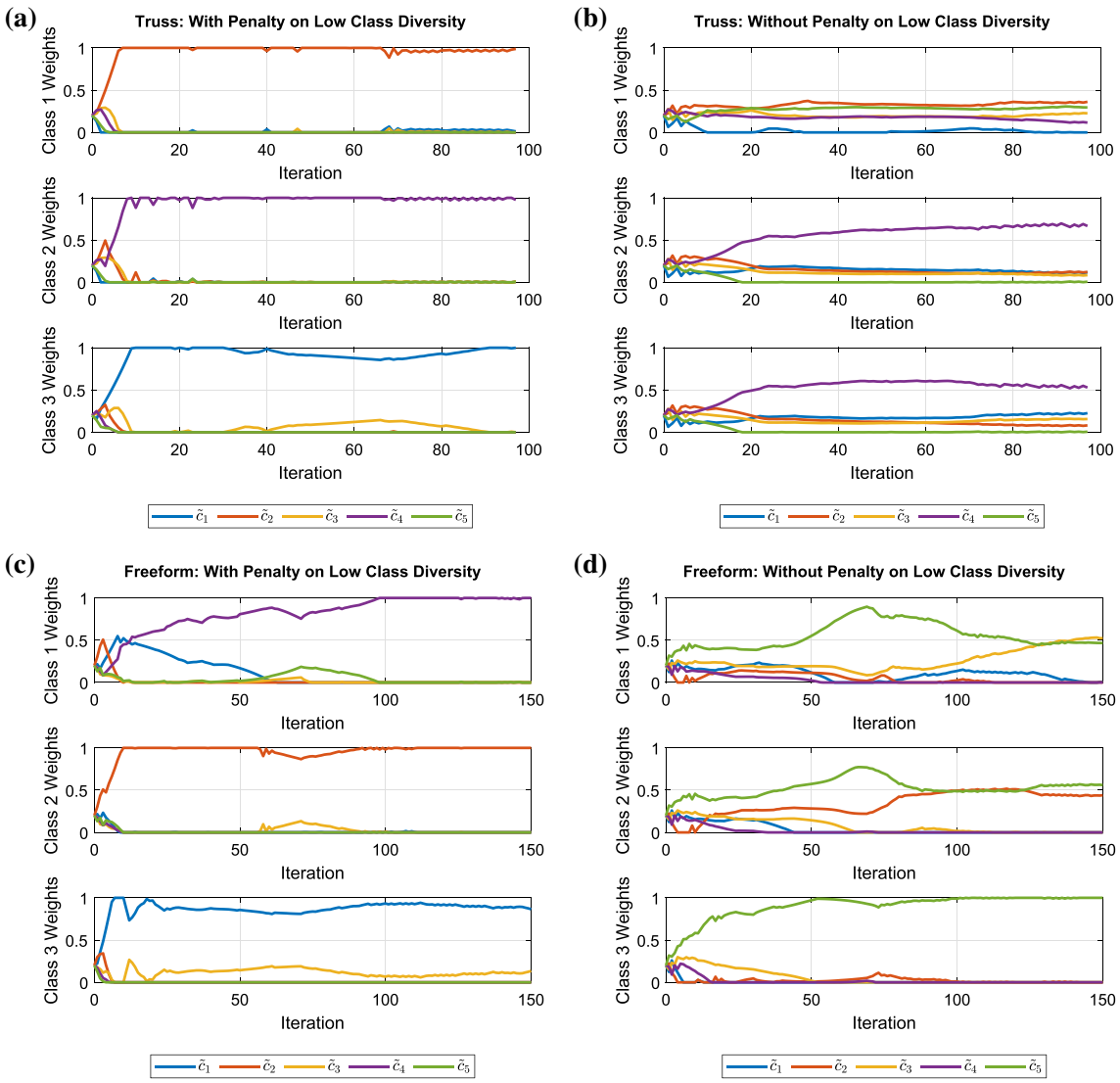


Fig. 16 For the 3-class MBB example, convergence plots of the class design variables with the proposed penalty (left column) and without (right column)

We also need the sensitivities of the radial filters. This follows the density filters of traditional TO methods closely (Sigmund 2007), and so we do not repeat it here for brevity.

For compliance (Eq. 11), the adjoint method (Bendsøe and Sigmund 2004) and chain rule allow us to derive the following with respect to the design variables:

$$\frac{\partial f_c}{\partial c_{e\cdot d}} = -\left[\sum_{n-1}^{N_{el}} \mathbf{u}_n^T \frac{\partial \mathbf{k}_n}{\partial \hat{c}_{e\cdot d}} \mathbf{u}_n\right] \frac{\partial \hat{c}_{e\cdot d}}{\partial c_{e\cdot d}}, \tag{C8}$$

$$\frac{\partial f_c}{\partial v_e} = -\left[\sum_{n-1}^{N_{el}} \mathbf{u}_n^T \frac{\partial \mathbf{k}_n}{\partial \hat{v}_e} \mathbf{u}_n\right] \frac{\partial \hat{v}_e}{\partial v_e}, \tag{C9}$$

and

$$\frac{\partial f_c}{\partial \xi_e^{(p)}} = -\left[\sum_{n-1}^{N_{el}} \mathbf{u}_n^T \frac{\partial \mathbf{k}_n}{\partial \hat{\xi}_e^{(p)}} \mathbf{u}_n\right] \frac{\hat{\xi}_e^{(p)}}{\xi_e^{(p)}}. \tag{C10}$$

The derivatives of the element effective stiffness matrices are

$$\frac{\partial \mathbf{k}_e}{\partial \hat{c}_{e\cdot d}} = \frac{\partial \mathbf{k}_e}{\partial \mathbf{C}_e} \frac{\partial \mathbf{C}_e}{\partial \hat{c}_{e\cdot d}}, \tag{C11}$$

$$\frac{\partial \mathbf{k}_e}{\partial \hat{v}_e} = \frac{\partial \mathbf{k}_e}{\partial \mathbf{C}_e} \frac{\partial \mathbf{C}_e}{\partial \hat{v}_e}, \tag{C12}$$

and

$$\frac{\partial \mathbf{k}_e}{\partial \hat{\xi}_e^{(p)}} = \frac{\partial \mathbf{k}_e}{\partial \mathbf{C}_e} \left[\sum_{d=1}^{D} \frac{\partial \mathbf{C}_e}{\partial \hat{c}_{e\cdot d}} \frac{\partial \hat{c}_{e\cdot d}}{\partial \hat{\xi}_e^{(p)}} \right]. \tag{C13}$$

Here, due to our data-driven framework, the gradients of the effective properties $\partial \mathbf{C}_e / \partial \hat{c}_{e\cdot d}$ and $\partial \mathbf{C}_e / \partial \hat{v}_e$ are obtained by backpropagating through the layers of the fully connected neural network (Hastie et al. 2009).

To obtain the sensitivities of the global volume fraction constraint, we can use Eq. C1 and the Sigmoid function (Eq. 9 in Sect. 2.3.2), which gives us a continuous approximation of a volume fraction ($\hat{v}_e^a$ from Eq. 10). First, the derivative of the Sigmoid applied to a microstructure's SDF ($\mathbf{\Phi}_e$) is

$$\frac{\partial S(\mathbf{\Phi}_e, \beta_1)}{\partial \mathbf{\Phi}_e} = \beta_1 S(\mathbf{\Phi}_e, \beta_1)\big(1 - S(\mathbf{\Phi}_e, \beta_1)\big). \tag{C14}$$

Therefore, the sensitivity of the global volume fraction can be decomposed as

$$\begin{aligned} \frac{\partial V_{\text{Global}}}{\partial \hat{c}_{e\cdot d}} &= \frac{\partial V_{\text{Global}}}{\partial \hat{v}_e^a} \frac{\partial \hat{v}_e^a}{\partial S} \frac{\partial S}{\partial \mathbf{\Phi}_e} \frac{\partial \mathbf{\Phi}_e}{\partial \hat{c}_{e\cdot d}}, \\ &= \frac{x_e}{N_{\text{el}}} \left(\frac{1}{n_{\text{el}}} \sum_{u=1}^{n_{\text{el}}} \frac{\partial S}{\partial \mathbf{\Phi}_e} \right) \frac{\partial \mathbf{\Phi}_e}{\partial \hat{c}_{e\cdot d}}, \end{aligned} \tag{C15}$$

and the sensitivities of the constraint itself with respect to the design variables are:

$$\frac{\partial g_1}{\partial c_{e\cdot d}} = \frac{1}{V^*_{\text{Global}}} \frac{\partial V_{\text{Global}}}{\partial \hat{c}_{e\cdot d}} \frac{\partial \hat{c}_{e\cdot d}}{\partial c_{e\cdot d}}, \tag{C16}$$

$$\frac{\partial g_1}{\partial v_e} = \frac{1}{V^*_{\text{Global}}}, \tag{C17}$$

and

$$\frac{\partial g_1}{\partial \xi_e^{(p)}} = \frac{1}{V^*_{\text{Global}}} \frac{\partial V_{\text{Global}}}{\partial \hat{c}_{e\cdot d}} \frac{\partial \hat{c}_{e\cdot d}}{\xi_e^{(p)}}. \tag{C18}$$

We note that these approximations of the microstructural volume fractions do introduce some error into the sensitivities but, in our experience, are minor.

Since we use the default algorithm for BESO developed by Huang and Xie (2007) to update $\mathbf{x}$, the filtering of the macroscale sensitivity numbers for the compliance problems can be found in the original paper. The only difference is that we use the effective stiffness predicted by the data-driven models. Thus, our sensitivity numbers are modified to be

$$\alpha_e = \mathbf{u}_e^T \mathbf{k}_e^H(\hat{\mathbf{c}}_e, \hat{v}_e)\mathbf{u}_e. \tag{C19}$$

For the low-diversity penalty function, the sensitivity can be derived using chain rule by taking the gradient of the log-determinant (Boyd and Vandenberghe 2004), then the Gaussian kernel and Euclidean distance between the class design variables, $\mathbf{c}^{(m)}$.

Finally, for the derivations of the first stage of the shape matching problem (Eq. 12), we refer the reader to our previous work (Wang et al. 2020a). For the second stage (Eq. 13) the derivations are straightforward to calculate by following the same steps above, substituting compliance for the mean squared error (MSE) between target and designed effective properties.

Acknowledgements We are grateful for the MMA codes from Prof. K. Svanberg at the Royal Institute of Technology, and the BESO codes from Profs. X. Huang and Y. M. Xie at RMIT University.

Funding The authors were supported by the National Science Foundation (NSF) CSSI program (Grant No. OAC-1835782). Yu-Chin Chan received funding from the NSF Graduate Research Fellowship (Grant No. DGE-1842165), and Liwei Wang from the Zhiyuan Honors Program for Graduate Students of Shanghai Jiao Tong University for his predoctoral visit at Northwestern University.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results All proposed schemes and algorithms are disclosed within this paper and developed by the authors in MATLAB (multiclass shape blending, concurrent design, data generation, deep learning) and Python (pre-processing of basis classes). Additional codes are adapted from open-source packages and are cited. The dataset from which the diverse freeform basis classes are extracted can be found at https://ideal.mech.northwestern.edu/.

References

Amir O, Aage N, Lazarov BS (2013) On multigrid-CG for efficient topology optimization. Struct Multidisc Optim 49(5):815–829. https://doi.org/10.1007/s00158-013-1015-5

Andreassen E, Andreasen CS (2014) How to determine composite material properties using numerical homogenization. Comput Mater Sci 83:488–495. https://doi.org/10.1016/j.commatsci.2013.09.006

Ba S, Myers WR, Brenneman WA (2015) Optimal sliced latin hypercube designs. Technometrics 57(4):479–487. https://doi.org/10.1080/00401706.2014.957867

Bendsøe MP, Sigmund O (2004) Topology optimization. Springer, Berlin, https://doi.org/10.1007/978-3-662-05086-6

Boley JW, Van Rees WM, Lissandrello C, Horenstein MN, Truby RL, Kotikian A, Lewis JA, Mahadevan L (2019) Shape-shifting

structured lattices via multimaterial 4d printing. Proc Natl Acad Sci 116(42):20856–20862. https://doi.org/10.1073/pnas.1908806116

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge. https://doi.org/10.1017/cbo9780511804441

Carballido-Gamio J, Bauer JS, Lee KY, Krause S, Majumdar S (2005) Combined image processing techniques for characterization of MRI cartilage of the knee. In: 2005 IEEE Engineering in Medicine and Biology 27th annual conference. IEEE, https://doi.org/10.1109/iembs.2005.1617116

Chan YC, Shintani K, Chen W (2019) Robust topology optimization of multi-material lattice structures under material and load uncertainties. Front Mech Eng 14(2):141–152. https://doi.org/10.1007/s11465-019-0531-4

Chan YC, Ahmed F, Wang L, Chen W (2020) METASET: exploring shape and property spaces for data-driven metamaterials design. J Mech Des. https://doi.org/10.1115/1.4048629

Christiansen AN, Nobel-Jørgensen M, Aage N, Sigmund O, Bærentzen JA (2013) Topology optimization using an explicit interface representation. Struct Multidisc Optim 49(3):387–399. https://doi.org/10.1007/s00158-013-0983-9

Chu S, Gao L, Xiao M, Zhang Y (2019) Multiscale topology optimization for coated structures with multifarious-microstructural infill. Struct Multidisc Optim 61(4):1473–1494. https://doi.org/10.1007/s00158-019-02428-y

Da D, Cui X, Long K, Cai Y, Li G (2019) Multiscale concurrent topology optimization of structures and microscopic multi-phase materials for thermal conductivity. Eng Comput 36(1):126–146. https://doi.org/10.1108/ec-01-2018-0007

Da D, Chan YC, Wang L, Chen W (2021) Data-driven and topological design of structural metamaterials for fracture resistance. Extreme Mech Lett. https://doi.org/10.1016/j.eml.2021.101528

Van Dijk NP, Maute K, Langelaar M, Van Keulen F (2013) Level-set methods for structural topology optimization: a review. Struct Multidisc Optim 48(3):437–472. https://doi.org/10.1007/s00158-013-0912-y

Du Z, Zhou XY, Picelli R, Kim HA (2018) Connecting microstructures for multiscale topology optimization with connectivity index constraints. J Mech Des 140:11. https://doi.org/10.1115/1.4041176

Eisenberger M, Lähner Z, Cremers D (2019) Divergence-free shape correspondence by deformation. Comput Gr Forum 38(5):1–12. https://doi.org/10.1111/cgf.13785

Garner E, Kolken HM, Wang CC, Zadpoor AA, Wu J (2019) Compatibility in microstructural optimization for additive manufacturing. Additiv Manuf 26:65–75. https://doi.org/10.1016/j.addma.2018.12.007

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning. Springer, New York. https://doi.org/10.1007/978-0-387-84858-7

Huang X, Xie Y (2007) Convergent and mesh-independent solutions for the bi-directional evolutionary structural optimization method. Finite Elements Anal Des 43(14):1039–1049. https://doi.org/10.1016/j.finel.2007.06.006

Jansen M, Pierard O (2020) A hybrid density/level set formulation for topology optimization of functionally graded lattice structures. Comput Struct 231(106):205. https://doi.org/10.1016/j.compstruc.2020.106205

Jia J, Da D, Hu J, Yin S (2021) Crashworthiness design of periodic cellular structures using topology optimization. Compos Struct 271(114):164. https://doi.org/10.1016/j.compstruct.2021.114164

Kumar S, Tan S, Zheng L, Kochmann DM (2020) Inverse-designed spinodoid metamaterials. npj Comput Mater. https://doi.org/10.1038/s41524-020-0341-6

Li D, Dai N, Tang Y, Dong G, Zhao YF (2019) Design and optimization of graded cellular structures with triply periodic level surface-based topological shapes. J Mech Des. https://doi.org/10.1115/1.4042617

Liu H, Zong H, Tian Y, Ma Q, Wang MY (2019) A novel subdomain level set method for structural topology optimization and its application in graded cellular structure design. Struct Multidisc Optim 60(6):2221–2247. https://doi.org/10.1007/s00158-019-02318-3

Lumpe TS, Shea K (2021) Computational design of 3d-printed active lattice structures for reversible shape morphing. J Mater Res. https://doi.org/10.1557/s43578-021-00225-2

Luo Y, Hu J, Liu S (2021) Self-connected multi-domain topology optimization of structures with multiple dissimilar microstructures. Struct Multidisc Optim 64(1):125–140. https://doi.org/10.1007/s00158-021-02865-8

Mirzaali MJ, Janbaz S, Strano M, Vergani L, Zadpoor AA (2018) Shape-matching soft mechanical metamaterials. Sci Rep. https://doi.org/10.1038/s41598-018-19381-3

Moré JJ (1978) The Levenberg-Marquardt algorithm: implementation and theory. In: More JJ (ed) Lecture notes in mathematics. Springer, Berlin, pp 105–116. https://doi.org/10.1007/bfb0067700

Oring A, Yakhini Z, Hel-Or Y (2021) Autoencoder image interpolation by shaping the latent space. In: Proceedings of the 38th international conference on machine learning, vol 139. PMLR, pp 8281–8290

Panesar A, Abdi M, Hickman D, Ashcroft I (2018) Strategies for functionally graded lattice structures derived using topology optimisation for additive manufacturing. Additive Manuf 19:81–94. https://doi.org/10.1016/j.addma.2017.11.008

Panetta J, Zhou Q, Malomo L, Pietroni N, Cignoni P, Zorin D (2015) Elastic textures for additive fabrication. ACM Trans Graph (TOG) 34(4):1–12. https://doi.org/10.1145/2766937

Ricci A (1973) A constructive geometry for computer graphics. Comput J 16(2):157–160. https://doi.org/10.1093/comjnl/16.2.157

Rohra AI, Kulkarni RK (2019) Survey on recent trends in image morphing techniques. In: 2019 International conference on smart systems and inventive technology (ICSSIT). IEEE, https://doi.org/10.1109/icssit46314.2019.8987879

Sanchez M (2015) Distance based heterogeneous volume modelling. PhD thesis, Bournemouth University

Sanders ED, Pereira A, Paulino GH (2021) Optimal and continuous multilattice embedding. Sci Adv 7(16):eabf4838. https://doi.org/10.1126/sciadv.abf4838

Schumacher C, Bickel B, Rys J, Marschner S, Daraio C, Gross M (2015) Microstructures to control elasticity in 3d printing. ACM Trans Graph (TOG) 34(4):1–13. https://doi.org/10.1145/2766926

scikit-fmm (2021) scikit-fmm: the fast marching method for python. https://github.com/scikit-fmm/scikit-fmm

Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidisc Optim 33(4–5):401–424. https://doi.org/10.1007/s00158-006-0087-x

Svanberg K (1987) The method of moving asymptotes-a new method for structural optimization. Int J Numer Methods Eng 24(2):359–373. https://doi.org/10.1002/nme.1620240207

Thomas AE, Guevelou S, Di Pasquale E, Chambard A, Duval JL, Chinesta F, Limousin V, Delgerie X, Leroy E (2020) Shape parametrization & morphing in sheet-metal forming. Procedia Manuf 47:702–706. https://doi.org/10.1016/j.promfg.2020.04.216

Wang L, Chan YC, Ahmed F, Liu Z, Zhu P, Chen W (2020a) Deep generative modeling for mechanistic-based learning and design of metamaterial systems. Comput Methods Appl Mech Eng 372(113):377. https://doi.org/10.1016/j.cma.2020.113377

Wang L, Chan YC, Liu Z, Zhu P, Chen W (2020b) Data-driven metamaterial design with laplace-beltrami spectrum as shape-DNA. Struct Multidisc Optim. https://doi.org/10.1007/s00158-020-02523-5

Wang L, Tao S, Zhu P, Chen W (2020c) Data-driven topology optimization with multiclass microstructures using latent variable gaussian process. J Mech Des. https://doi.org/10.1115/1.4048628

Wang Y, Zhang L, Daynes S, Zhang H, Feih S, Wang MY (2018) Design of graded lattice structure with optimized mesostructures for additive manufacturing. Mater Des 142:114–123. https://doi.org/10.1016/j.matdes.2018.01.011

Xia L, Breitkopf P (2014) Concurrent topology optimization design of material and structure within fe2 nonlinear multiscale analysis framework. Comput Methods Appl Mech Eng 278:524–542. https://doi.org/10.1016/j.cma.2014.05.022

Xiao M, Liu X, Zhang Y, Gao L, Gao J, Chu S (2021) Design of graded lattice sandwich structures by multiscale topology optimization. Comput Methods Appl Mech Eng 384(113):949. https://doi.org/10.1016/j.cma.2021.113949

Xu L, Cheng G (2018) Two-scale concurrent topology optimization with multiple micro materials based on principal stress orientation. Struct Multidisc Optim 57(5):2093–2107. https://doi.org/10.1007/s00158-018-1916-4

Zhang W, Yuan J, Zhang J, Guo X (2015) A new topology optimization approach based on moving morphable components (MMC) and the ersatz material model. Struct Multidisc Optim 53(6):1243–1260. https://doi.org/10.1007/s00158-015-1372-3

Zhang Y, Li H, Xiao M, Gao L, Chu S, Zhang J (2018) Concurrent topology optimization for cellular structures with nonuniform microstructures based on the kriging metamodel. Struct Multidisc Optim 59(4):1273–1299. https://doi.org/10.1007/s00158-018-2130-0

Zhou XY, Du Z, Kim HA (2019) A level set shape metamorphosis with mechanical constraints for geometrically graded microstructures. Struct Multidisc Optim 60(1):1–16. https://doi.org/10.1007/s00158-019-02293-9

Zobaer SMT, Sutradhar A (2020) An energy-based method for interface connectivity of incompatible microstructures through parametric modeling. Comput Methods Appl Mech Eng 370(113):278. https://doi.org/10.1016/j.cma.2020.113278