Set-In-Stone: Worst-Case Optimization of Structures Weak in Tension

CHRISTIAN SCHUMACHER*, Disney Research
JONAS ZEHNDER*, Disney Research and Université de Montréal
MORITZ BÄCHER, Disney Research

Fig. 1. We optimize the strength-to-weight ratio and mass distribution of binder-jetted, large-scale structures under worst-case loads, in order to make them durable (left, middle, right) and stand (right). Given a combination of live, wind, dead, and thermal loads common in structural engineering, our optimization identifies worst-case wind directions and contact regions, minimizing resulting peak stresses.

Large-scale binder jetting provides a promising alternative to manual sculpting of sandstone. The weak build material, however, severely limits its use in architectural ornamentation. We propose a structural optimization that jointly optimizes an ornament's strength-to-weight ratio and balance under self-weight, thermal, wind, and live loads. To account for the difference in the tensile and compressive strength of the build material, we turn the Bresler-Pister criterion into a failure potential, measuring the distance to failure. Integrated into an XFEM-based level set formulation, we minimize this potential by changing the topology and shape of the internal structure. To deal with uncertainties in the location of live loads, and the direction of wind loads, we first estimate loads that lead to the weakest structure, then minimize the potential of failure under identified worst-case loads. With the help of first-order optimality constraints, we unify our worst-case load estimation and structural optimization into a *continuous* optimization. We demonstrate applications in art, furniture design, and architectural ornamentation with three large-scale 3D printed examples.

CCS Concepts: • **Computing methodologies** → **Physical simulation**; • **Applied computing** → *Computer-aided design*;

Additional Key Words and Phrases: Structural optimization, XFEM, worst-case load, Bresler-Pister failure criterion

*The first two authors contributed equally.

Authors' addresses: Christian Schumacher, Disney Research; Jonas Zehnder, Disney Research and Université de Montréal; Moritz Bächer, Disney Research.

0730-0301/2018/11-ART252 $15.00
https://doi.org/10.1145/3272127.3275085

ACM Reference Format:
Christian Schumacher, Jonas Zehnder, and Moritz Bächer. 2018. Set-In-Stone: Worst-Case Optimization of Structures Weak in Tension. *ACM Trans. Graph.* 37, 6, Article 252 (November 2018), 13 pages. https://doi.org/10.1145/3272127.3275085

1 INTRODUCTION

For centuries, architectural ornaments have seen widespread use as embellishments on buildings. Among sculptors, standstone is a preferred material. Its distinct grain makes the stone easy to carve, and facilitates the sculpting of very fine detail. However, manual sculpting requires skill, is time consuming, and expensive.

Large-scale binder jetting provides a promising alternative. Voxeljet's VX4000 [Voxeljet 2018], for instance, has a build tray size of $4 \times 2 \times 1$ m and jets at a stunning resolution of 300 dpi with a layer thickness of 300 microns. Binder jetting of concrete is possible [D-Shape 2018] but has yet to mature, and printers with even larger build volumes are becoming available.

A weakness of binder jetting is the strength of its build material. Similar to concrete, printed structures are far weaker under tension than under compression. This asymmetry in strength is the reason why concrete is traditionally reinforced with a grid of steel bars.

Analyzing structures under live loads, self-weight, and wind, we observe that this asymmetry in strength severely limits the use of binder jetting at the scale of available build tray sizes. Solid models as well as models hollowed to constant thickness would break under moderate loads. Hence, a structural optimization that accounts for this asymmetry is key to making use of this technology.

In this paper, we propose an algorithm to jointly optimize an ornament's strength-to-weight ratio and balance, under loads with parameterized uncertainties. To account for the asymmetry in strength,

we recast the Bresler-Pister criterion as a failure potential, measuring the *distance* to failure. Integrated into a mature XFEM-based level-set formulation [Noël et al. 2015] that handles topological changes and avoids remeshing, we then minimize this failure potential by making changes to the ornament's internal structure.

Freely standing in a park, sculptures are often climbed upon. Because it is not known where a person would stand on or hold onto a printed ornament, there is uncertainty in the so-called live loads. A printed gargoyle, mounted on a cathedral, is exposed to severe weather conditions. While a structural engineer can provide us with a value for the expected wind magnitude, the direction of wind loads is again unknown. Parameterizing these uncertainties, we propose to nest

- a structural analysis to ensure feasibility of a design,
- a worst-case load estimation to identify the loads leading to the weakest structure,
- and a structural or design optimization that adjusts our level set parameters to minimize the failure potential under identified worst-case loads

into a *unified continuous* optimization. Using first-optimality constraints on our analysis (inner loop) and load estimation (mid-loop), we can compute *analytical gradients* for our design optimization (outermost loop).

We demonstrate our technique on three printed examples, illustrating applications in art, furniture design, besides architectural ornamentation (see Fig. 1): we optimize a Gargoyle to be structurally sound under worst-case wind loads (left), a stool to withstand unknown live loads (middle), and a statue to stably stand (right). Performing mechanical testing on bridge designs that are optimized with the von Mises and the Bresler-Pister criterion, we validate our failure potential.

Contributions. In summary, we contribute

- a strategy to turn failure criteria, including ones that account for differences in tensile and compressive strengths, into failure potentials well-suited for optimization
- a nesting strategy to unify the analysis, a worst-case load estimation, and a structural optimization into a continuous optimization, interfacing with *arbitrary* parameterized loads
- "softened" Dirichlet conditions to avoid unrealistically high stresses close to mount locations

While we implement a specific instance of this nesting strategy, we see applications beyond our structural optimization.

2 RELATED WORK

Triggered by advances in manufacturing technologies, fabrication-oriented design has received increasing attention in the computer graphics community [Bermano et al. 2017]. We focus our literature review on the design of 3D printed parts whose strength-to-weight ratio and mass distributions were optimized. We first review prior art in graphics, followed by a discussion of closely related work in structural, shape, and topology optimization. We conclude with a review of worst-case approaches.

Strength and Mass Distribution Optimization. To improve the structural strength of printed parts, methods have been devised to optimize print orientation [Umetani and Schmidt 2013], add struts in the interior [Stava et al. 2012; Wang et al. 2013], or generate Voronoi-based internal structures [Lu et al. 2014]. Related to our work is Prévost et al. [2013] who optimize the internal mass distribution of printed parts to make them stand stably. Follow-up work includes dynamic balancing [Bächer et al. 2014] and fully submerged and floating objects [Musialski et al. 2015]. In contrast, we optimize a combination of structural objectives and mass property targets, relying on a level set discretization to allow for topological changes.

Shape and Topology Optimization. An exhaustive review of prior art in this domain is beyond the scope of this paper and we refer the interested reader to standard books [Bendsøe and Sigmund 2004; Sokolowski and Zolesio 1992] and an excellent survey [Deaton and Grandhi 2014] on the subject matter. Here, we look at a selection of works from structural optimization.

Based on traditional topology optimization, Wu et al. [2016] propose a GPU-friendly multi-grid approach that makes topology optimization more scalable, enabling resolutions of several million elements. Subsequent work introduces infill optimization [Wu et al. 2018] that generates structures robust to material defects or load perturbations. Other approaches optimize the shape of 3d printed parts [Zhou et al. 2016] or the thickness of shells [Zhao et al. 2017].

We base our discretization on work by Noël et al. [2015], using extended finite elements with a ridge enrichment to compute shape derivatives and evolve the material interface with a level set discretization [Osher and Fedkiw 2003]. Other related approaches [Allaire et al. 2009, 2004; Wang et al. 2003] use level set propagation for structural optimization.

Stress Minimization. The overwhelming majority of approaches rely on the von Mises criterion to minimize or bound stresses (see, e.g., [Panetta et al. 2017]). However, due to the asymmetry in the tensile and compressive strength of binder-jetted material, the use of the von Mises criterion is suboptimal. Departing from this criterion, Luo et al. [2012] use the Drucker-Prager criterion to formulate stress constraints. In contrast, we base our formulation on the Bresler-Pfister criterion, introducing an approach to derive distance measures for general failure criteria well-suited for optimization.

A popular approach to minimize the potential of failure introduces stress bounds on a per-element level (see, e.g., [Bendsøe and Sigmund 2004]). However, while methods that aggregate local constraints into regions exist [Deaton and Grandhi 2014] , they tend to be expensive due to the high number of constraints. We propose to minimize the maximum failure potential among all elements by measuring the *distance to failure* and implementing a smooth max.

Worst-Case Optimization. The solution of a structural optimization largely depends on the applied loads. Most techniques assume these loads to be well-defined. However, in practical applications loads are subject to uncertainties. Addressing robustness under small perturbations, traditional compliance optimization was first extended [Cherkaev and Cherkaeva 1999; de Gournay et al. 2008], followed by work that uses a level set discretization for stress optimization instead [Allaire et al. 2009].

Modeling more general loads, Zhou et al. [2013] formulate a worst-case *analysis* under unknown loads that act normal to the surface.

Table 1. Stress quantities.

Cauchy stress $\boldsymbol{\sigma}$			deviatoric stress $\mathbf{s} = \boldsymbol{\sigma} - \sigma_m \mathbf{I}$		
$(\sigma_1, \sigma_2, \sigma_3)$			(s_1, s_2, s_3)		
I_1	=	$\operatorname{tr}(\boldsymbol{\sigma})$	J_1	=	$\operatorname{tr}(\mathbf{s})$
	=	$\sigma_1 + \sigma_2 + \sigma_3$		=	$s_1 + s_2 + s_3 = 0$
I_2	=	$\frac{1}{2}\left[\operatorname{tr}(\boldsymbol{\sigma})^2 + \operatorname{tr}(\boldsymbol{\sigma}^2)\right]$	J_2	=	$\frac{1}{2}\operatorname{tr}(\mathbf{s}^2) = \frac{1}{3}I_1^2 - I_2$
	=	$\sigma_1\sigma_2 + \sigma_2\sigma_3 + \sigma_3\sigma_1$		=	$\frac{1}{2}\left(s_1^2 + s_2^2 + s_3^2\right)$
I_3	=	$\det(\boldsymbol{\sigma})$	J_3	=	$\det(\mathbf{s})$
	=	$\sigma_1\sigma_2\sigma_3$		=	$s_1 s_2 s_3$

Panetta et al. [2017] introduce a worst-case formulation to relieve stress in microstructures under all unit loads. To optimize printed parts to withstand unknown interactions, Ulu et al. [2017] assume loads to act in negative normal direction. In contrast, we nest an analysis, a worst-case load estimation, and a structural optimization to interface with *arbitrary* loads, thereby minimizing a structure's potential of failure under uncertainties spanned by their parameters.

Langlois et al. [2016] use rigid body simulation to predict failure modes of objects. Based on this measure, they formulate a stress-constrained topology optimization. A simulation-based approach could help with extending our set of supported worst-case loads.

3 BALANCING STRENGTH AND MASS

Our technique takes as input a 3D model that fits onto a build tray of a targeted large-scale printer (see Fig. 2 Input). Besides the 3D model, the user provides strength and material parameters, specifies a set of unparameterized (self-weight, thermal loads) and parameterized loads (wind, live loads), and labels a subset of the surface as fixed to the ground or to a mounting system. Putting users in control, we ask them to specify the design space, limiting changes to the interior structure to respective regions.

Given the input, we first estimate the loads that lead to the weakest structure (Worst-Case Loads, Sec. 5). Whenever we update load parameters, we balance the internal forces with the applied load (Simulation, Sec. 4). Given a set of worst-case loads, we then minimize the potential of failure by changing the shape and topology of the internal structure (Design Optimization, Sec. 6).

In the remainder of this section, we discuss failure criteria and introduce our failure potential. We then discuss our nesting strategy to formulate a unified, continuous worst-case optimization.

3.1 Failure Criteria

To analyze the soundness of a structure, structural engineers rely on failure criteria [Chen and Saleeb 1994]: a structure reaches its point of failure if it cannot carry any further load. Failure criteria are functions of the state of stress $\boldsymbol{\sigma}$ and are parameterized with material-dependent strength values where strength marks the limit state of stress beyond which a structure fractures.

To better understand failure criteria, we first shed light onto several stress quantities (see Tab. 1). From the Cauchy stress $\boldsymbol{\sigma}$, we can extract principal stresses σ_1, σ_2, and σ_3. While the Cauchy stress itself depends on the chosen coordinate frame, failure criteria are best defined on quantities that are independent of a particular frame. The invariants I_1, I_2, and I_3 lend themselves, where $I_1 = \sigma_1 + \sigma_2 + \sigma_3$ is responsible for volume changes. Scaling I_1 by a factor, we form the *mean* or *pure hydrostatic stress* $\sigma_m = \frac{1}{3}I_1$.

The pure hydrostatic stress, in turn, allows us to decompose the Cauchy stress into one component causing pure shear and one causing the volume to change: $\boldsymbol{\sigma} = \mathbf{s} + \sigma_m \mathbf{I}$. Referred to as *deviatoric stress*, the tensor $\mathbf{s}$ shares the principal directions with $\boldsymbol{\sigma}$ while its principal stresses $s_i = \sigma_i - \sigma_m$ are shifted by the mean stress. By construction, the first invariant $J_1 = s_1 + s_2 + s_3$ is zero, thus confirming that $\mathbf{s}$ does not cause the volume to change. Most relevant in our context is the second invariant J_2 of pure shear.

The separation into a pressure and a shear component provides us with geometric insight (compare with Fig. 3 top, left): for a given stress tensor $\boldsymbol{\sigma}$ with principal stresses $(\sigma_1, \sigma_2, \sigma_3)$, we can compute the cylindrical Haigh-Westergaard coordinates (ξ, ρ, θ)

$$\xi = \frac{1}{\sqrt{3}} I_1 \qquad \rho = \sqrt{2J_2} \qquad \theta = \arccos\left(\frac{\sqrt{3}}{2}\frac{s_1}{\sqrt{J_2}}\right).$$

All stress points $\boldsymbol{\sigma}$ with equal invariant I_1 lie on a plane whose normal points in the direction of the hydrostatic pressure axis $\sigma_1 = \sigma_2 = \sigma_3$, offset by the hydrostatic coordinate ξ. The deviatoric coordinate ρ and similarity angle θ can be understood as polar coordinates of $\boldsymbol{\sigma}$ within this deviatoric plane.

Failure surfaces separate stresses causing failure from unproblematic ones and are best understood when we work with these cylindrical coordinates. For isotropic materials, the labels attached to the principal axes σ_i are arbitrary. Hence, the intersection of the failure surface with a hydrostatic plane ξ has to be threefold rotationally symmetric as we illustrate in Fig. 3 top, right. Failure surfaces are further convex and smooth [Chen and Saleeb 1994].

Most common in graphics is the von Mises criterion with an infinite cylinder of radius $\rho = \sqrt{2}A$ as failure surface (see Fig. 3 bottom, left). The constant $A(\sigma)$ depends on a uniaxial strength parameter σ. We therefore call this criterion a one-parameter model.

However, parameterized with a single parameter, the von Mises criterion assumes the *tensile strength* σ_t and *compressive strength* σ_c to be equal. Seeking a criterion that accounts for asymmetries in strength, the two-parameter Drucker-Prager criterion with failure surface $\rho(\xi) = \sqrt{2}A + \sqrt{5}B\xi$ is a candidate (see Fig. 3 bottom, middle). Due to the linear dependence of radius ρ on ξ, the failure surface is an open cone whose slope and origin is controlled by the two constant coefficients $A(\sigma_t, \sigma_c)$ and $B(\sigma_t, \sigma_c)$.

Accounting for the asymmetry in strength, the Drucker-Prager criterion is nonetheless modeling our build material to an insufficient degree: while tensile stresses are bounded (see tip of cone in Fig. 3), compressive stresses can grow infinitively high while remaining within the cone. Because no structure can withstand infinitely high stresses, this criterion is too coarse of a model, and the unboundedness makes this candidate problematic for use in structural optimization.

A three-parameter criterion that has a closed failure surface and better approximates the behavior of our build material is the Bresler-Pister criterion $\rho(\xi) = \sqrt{2}A + \sqrt{5}B\xi + 3\sqrt{2}C\xi^2$. As we illustrate in Fig. 4 bottom, right, the criterion implements a parabolic dependence of radius ρ on ξ. The coefficients $A(\sigma_t, \sigma_c, \sigma_b)$, $B(\sigma_t, \sigma_c, \sigma_b)$, and $C(\sigma_t, \sigma_c, \sigma_b)$ are parameterized with the *biaxial compressive strength* σ_b besides the tensile and compressive strength.

Fig. 2. **Overview.** As input, our optimization takes a 3D model, parameterized loads, grounded surface regions, and a design space (Input). To optimize the strength-to-weight ratio, we first solve for the loads that maximize the potential of failure (Worst-Case Loads) under static equilibrium constraints (Simulation). We then minimize the potential of failure under these worst-case loads (Design Optimization), adding objectives to reduce weight and place the center of mass. To unify all three stages into a continuous optimization, we enforce first-optimality of our worst-case load estimation and simulation when solving our design optimization. Our output structure is sound and withstands worst-case loads drawn from a subspace of uncertainty (Output).

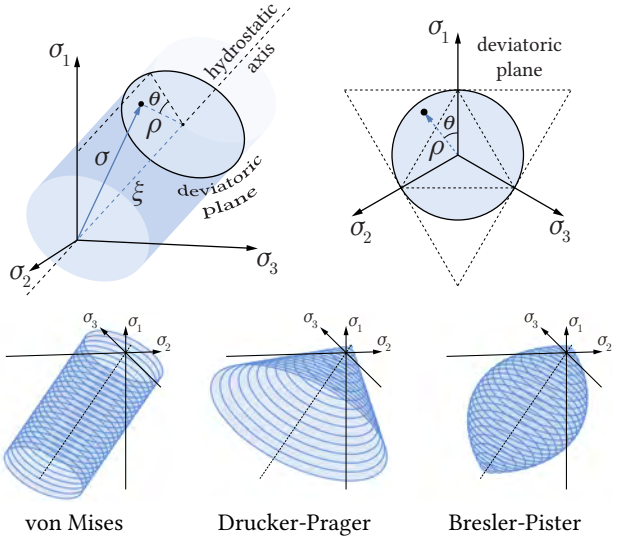

Fig. 3. **Failure Surfaces.** Stress point in cylindrical Haigh-Westergaard coordinates (top, left). Intersection curves of surfaces with deviatoric planes are threefold symmetric, smooth, and convex (top, right). The convexity limits the curves to the three dotted triangles. Failure surfaces (bottom) for the von Mises (infinite cylinder, left), the Drucker-Prager criterion (open cone, middle), and the Bresler-Pister criterion (right).

All failure criteria we have discussed so far are independent of the parameter θ. The intersections of the failure surface with the deviatoric planes are therefore circular. For concrete-like materials, these intersection curves are noncircular in general, changing from a nearly triangular shape for tensile stresses to a more circular shape for high compressive stresses. While four- or five-parameter criteria model these subtleties [Chen and Saleeb 1994], advanced mechanical testing equipment is needed to identify the required strength parameters. While our recasting of criteria as distance metrics (see next section) is generally applicable, the Bresler-Pister criterion represents a good compromise between model accuracy, practicality when it comes to characterization of its parameters (see Sec. 7), and suitability for numerical optimization. Note that the coefficients A, B, and C vary between criteria as we discuss in detail in our supplemental material.

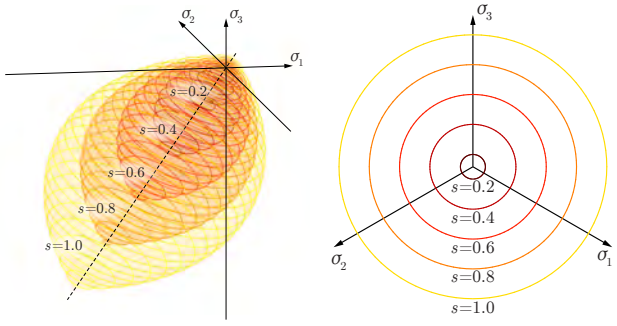

Fig. 4. **Failure Potential.** Bresler-Pister failure surfaces for several scale factor values s (left). Intersection curves with the deviatoric plane (right) motivate the use of s as a measure of distance to failure.

3.2 Failure Potential

Our ultimate goal is to define an objective to minimize a structure's potential of failure. However, while criteria allow us to evaluate if a stress point $\boldsymbol{\sigma}$ lies on the failure surface or not, they do not provide a means to measure the *distance* to failure. To turn criteria into distance metrics, we propose to multiply the strength parameters of their constant coefficients with a uniform scale factor s, then solving for the latter. Illustrating our recasting on the Bresler-Pister criterion, we refer the interested reader to our supplemental material for detailed derivations for other criteria:

Scaling the strength parameters σ_t, σ_c, σ_b in the coefficients A, B, C of the Bresler-Pister criterion with factor s, we observe that we can express them with coefficients with unscaled parameters $A(s\sigma_t, s\sigma_c, s\sigma_b) = sA(\sigma_t, \sigma_c, \sigma_b)$, $B(s\sigma_t, s\sigma_c, s\sigma_b) = B(\sigma_t, \sigma_c, \sigma_b)$, and $C(s\sigma_t, s\sigma_c, s\sigma_b) = \frac{1}{s}C(\sigma_t, \sigma_c, \sigma_b)$, pointing the reader to the Appendix for coefficient formulas. Plugged into the variant $\sqrt{J_2} - A - BI_1 - CI_1^2 = 0$ of the Bresler-Pister criterion that is well-suited for optimization (stress invariants can be computed with simple formulas, see Tab. 1), we get a quadratic function we can solve for the scale factor

$$s(\boldsymbol{\sigma}) = \frac{\sqrt{J_2} - BI_1 + \sqrt{(B^2 - 4AC)I_1^2 - 2BI_1\sqrt{J_2} + J_2}}{2A},$$

ignoring the negative solution.

As we illustrate in Fig. 4 right, the scale factor s provides us with a distance metric that fulfills our desiderata, measuring the *potential of failure*: it returns zero if the Cauchy stress is zero, taking on increasing values in the interval $[0, 1]$, the closer $\boldsymbol{\sigma}$ is to the failure surface. For stresses on the failure surface, $s(\boldsymbol{\sigma})$ evaluates to 1. If the Cauchy stress grows beyond failure, s monotonically increases with the distance of $\boldsymbol{\sigma}$ to the failure surface.

Our goal is therefore to ensure that the distance-to-failure metric s is below 1 at *all* locations within a structure under *all* expected loads.

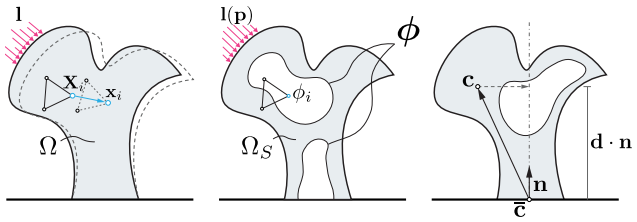

Fig. 5. **Worst-Case Optimization** To analyze structures, we discretize the design domain Ω into finite elements (left): under a load $\mathbf{l}$, vertices at rest $\mathbf{X}_i$ deform to positions $\mathbf{x}_i$. To optimize their strength-to-weight ratio, we parameterize uncertainties in load cases $\mathbf{l}(\mathbf{p})$ and represent the structure's volume $\Omega_s \subset \Omega$ with a discretized level-set function $\boldsymbol{\phi}$ (middle). To control the center of mass $\mathbf{c}$, the difference between its horizontal coordinate and its desired projected location $\bar{\mathbf{c}}$ on the ground plane is minimized.

3.3 Worst-Case Optimization

Discretized into finite elements (compare with Fig. 5 left) with n vertices, we can analyze if a structure is sound under a load case $\mathbf{l} \in \mathbb{R}^{3n}$ by minimizing the simulation objective

$$f_{\text{sim}}(\mathbf{x}) = W(\mathbf{X}, \mathbf{x}) - \mathbf{l}^T(\mathbf{x} - \mathbf{X}) \tag{1}$$

to static equilibrium. The internal energy W is material-dependent and defines the response of the structure to work performed by $\mathbf{l}$ (see Sec. 4). At the equilibrium, internal forces $\frac{\partial}{\partial \mathbf{x}} W(\mathbf{X}, \mathbf{x})$ are in balance with the applied load, and the vertices of the elements at rest $\mathbf{X} \in \mathbb{R}^{3n}$ deform to the configuration $\mathbf{x} \in \mathbb{R}^{3n}$. In other words, we can solve for the deformed configuration $\mathbf{x}$ by minimizing our simulation objective to first-order optimality $g_{\text{sim}}(\mathbf{x}) = 0$. After analysis, we can evaluate the Cauchy stress at any location $\mathbf{X} \in \Omega$, and check if the failure potential s is below 1.

While this is sufficient for analysis under clearly defined loads (e.g., self-weight), live or wind loads bear uncertainty in their locations or direction. Parameterizing this uncertainty with a set of parameters $\mathbf{p}$ (see Fig. 5 middle and Sec. 5), we aim at analyzing a structure under *all* loads $\mathbf{l}(\mathbf{p})$. Hence, to see whether a structure is sound, we seek to identify the parameters that lead to the maximum potential of failure within the domain Ω_s. While we could use a smooth maximum function to approximate the max-operator, we use an integral over the exponentiated distance-to-failure instead

$$f_{\text{fail}}(\mathbf{p}) = \int_{\Omega_s} [s(\mathbf{X}, \mathbf{x}(\mathbf{p}))]^\gamma \, dV, \tag{2}$$

penalizing large values of s exponentially more with exponent $\gamma > 2$. The higher the value of γ, the more we penalize peak stresses.

Whenever we adjust the parameters, the corresponding load case will change, and hence also the deformed configuration. To identify the parameters $\mathbf{p}$ that lead to the *weakest* structure, we maximize the failure potential over an equilibrium constraint

$$\max_{\mathbf{p}} f_{\text{fail}}(\mathbf{p}) \quad \text{s.t.} \quad g_{\text{sim}}(\mathbf{p}, \mathbf{x}(\mathbf{p})) = 0. \tag{3}$$

As we will describe in Sec. 5, we implicitly enforce the first-order optimality of our simulation objective, minimizing $-f_{\text{fail}}$ with an analytical gradient that we compute with the help of the implicit function theorem.

Our ultimate goal is to optimize the strength-to-weight ratio of a structure under identified worst-case loads. To be able to change a structure's shape and topology, we rely on a level-set formulation [Noël et al. 2015] that uses extended finite elements (XFEM) to avoid expensive remeshing. Postponing a detailed discussion to Secs. 4 and 6, we hereafter assume the domain Ω_s to be parameterized with a discretized level-set function $\boldsymbol{\phi} \in \mathbb{R}^n$ (see Fig. 5 middle).

Observing that adjustments to $\boldsymbol{\phi}$ cause worst-case loads to move, and, in turn, the deformed configuration to change, we aim at identifying the best trade-off between the potential of failure and the weight or volume V of the structure

$$f_{\text{design}}(\boldsymbol{\phi}) = w_{\text{fail}} f_{\text{fail}}(\mathbf{p}(\boldsymbol{\phi})) + w_{\text{weight}} V(\Omega_s(\boldsymbol{\phi})). \tag{4}$$

Due to the continuous coupling between $\boldsymbol{\phi}$, $\mathbf{p}$, and $\mathbf{x}$, we seek to formulate a unified, continuous worst-case optimization. To this end, we propose to *nest* simulation, worst-case load estimation, and structural optimization with first-order optimality constraints

$$\begin{aligned} &\min_{\boldsymbol{\phi}} f_{\text{design}}(\boldsymbol{\phi}) \\ &\text{s.t.} \quad g_{\text{fail}}(\boldsymbol{\phi}, \mathbf{p}(\boldsymbol{\phi}), \mathbf{x}(\boldsymbol{\phi}, \mathbf{p}(\boldsymbol{\phi}))) = 0 \\ &\qquad g_{\text{sim}}(\boldsymbol{\phi}, \mathbf{p}(\boldsymbol{\phi}), \mathbf{x}(\boldsymbol{\phi}, \mathbf{p}(\boldsymbol{\phi}))) = 0. \end{aligned} \tag{5}$$

Solving our simulations and worst-case load estimation to first-order optimality whenever we evaluate our design objective or its gradient, we can compute analytical gradients for the unified problem (see Sec. 6).

Optionally, the user may control the placement of the center of mass $\mathbf{c}$. This could help to make structures stably stand or, if fixed to the ground, reduce stresses close to their mounting locations. To this end, we let the user specify the desired location of the projected center $\bar{\mathbf{c}}$, defining a line through $\bar{\mathbf{c}}$ in direction of the unit normal $\mathbf{n}$ of the ground plane (see Fig. 5 right). In design optimizations, we then add the objective

$$f_{\text{CoM}}(\boldsymbol{\phi}) = \frac{1}{2} \| (\mathbf{d}(\boldsymbol{\phi}) \cdot \mathbf{n})\, \mathbf{n} - \mathbf{d}(\boldsymbol{\phi}) \|^2 \text{ with } \mathbf{d}(\boldsymbol{\phi}) = \mathbf{c}(\Omega_s(\boldsymbol{\phi})) - \bar{\mathbf{c}}, \tag{6}$$

penalizing the distance between the horizontal component of $\mathbf{c}$ and our target $\bar{\mathbf{c}}$. We use a weight w_{CoM} to control its influence.

To make our weights independent of the dimensions of our input, we divide the failure and weight objectives by the volume of the filled structure Ω, and our center of mass objective by its cube root. To optimize structures under several unparameterized and parameterized loads $\mathbf{l}^k$, we simulate to equilibria $\mathbf{x}^k$ when minimizing the maximum potential of failure among all loads k.

4 SIMULATION

Our structures undergo only small deformations and we base our formulation on the infinitesimal strain theory. Largely following Noël et al. [2015], we keep the description of our analysis brief, discussing our softened Dirichlet conditions in more detail.

We discretize the volume Ω enclosed by the input model with tetrahedral elements and interpolate the undeformed nodes $\mathbf{X}_i$ and level set values ϕ_i within the elements with standard linear shape functions $N_i(\boldsymbol{\xi})$. To interpolate deformed nodes $\mathbf{x}_i$, we differentiate between uncut (incident ϕ_i have the same sign) and cut elements (ϕ_i have different signs), relying on standard interpolation for uncut elements. For cut elements, we use an enrichment

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{i=1}^{4} \mathbf{x}_i N_i(\boldsymbol{\xi}) + \sum_{i=1}^{4} \hat{\mathbf{x}}_i \hat{N}_i(\boldsymbol{\xi}) \quad \text{with} \quad \hat{N}_i(\boldsymbol{\xi}) = N_i(\boldsymbol{\xi})\psi(\boldsymbol{\xi}),$$

interpolating additional degrees of freedom $\hat{\mathbf{x}}_i$ with shape functions $\hat{N}_i(\boldsymbol{\xi})$. To represent the discontinuity at the interface, we use a ridge function [Moës et al. 2003]

$$\psi(\boldsymbol{\xi}) = \sum_{i=1}^{4} |\phi_i| \, N_i(\boldsymbol{\xi}) - \left| \sum_{i=1}^{4} \phi_i N_i(\boldsymbol{\xi}) \right|.$$

Note that the deformation gradient $\mathbf{F}$ is *not* constant for enriched elements, and numerical integration is required to integrate the linear strain energy density

$$\Psi(\mathbf{F}(\boldsymbol{\xi})) = \mu \boldsymbol{\varepsilon}(\boldsymbol{\xi}) : \boldsymbol{\varepsilon}(\boldsymbol{\xi}) + \frac{\lambda}{2} \mathrm{tr}^2(\boldsymbol{\varepsilon}(\boldsymbol{\xi})) \quad \text{with} \quad \mathbf{F}(\boldsymbol{\xi}) = \frac{\partial \mathbf{x}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \left(\frac{\partial \mathbf{X}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right)^{-1},$$

where $\boldsymbol{\varepsilon}$ is the Cauchy strain and μ and λ are the Lamé constants. Due to the discontinuity at the interface, we cannot apply a quadrature rule to a cut element directly. We follow common practice and decompose cut elements e into isoparametric sub-elements $\mathcal{E}_s$ that conform to the interface

$$W_e = \sum_s \int_{\mathcal{E}_s} \Psi(\boldsymbol{\xi}) \det \left[\frac{\partial \mathbf{X}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right] d\boldsymbol{\xi}.$$

With the determinant of the Jacobian of the mapping $\mathbf{X}(\boldsymbol{\xi})$, we account for the change of variables to elemental coordinates. Uncut elements are analytically integrated: $W_e = V_e \Psi_e$ where V_e denotes the rest-volume of the element and Ψ_e the strain energy density for its constant deformation gradient.

We represent the interface with a *weak* discontinuity and use an *ersatz material* [Allaire et al. 2004] for elements and sub-elements in the void space: we multiply the Young's modulus for integration points outside of Ω_s with a scale factor (we use 10^{-4} for our demonstrations). While we observe deformations of the void space to be negligible, we can expect an increase in performance if we model the interface with a *strong* discontinuity (see, e.g., [Van Miegroet and Duysinx 2007]), removing the degrees of freedom for void (sub-) elements from the equation system.

4.1 Softening Dirichlet Conditions

Rubber and foam is often used as shim material between a mounting system and a structure. Hence, Dirichlet conditions, while straightforward to enforce, lead to unrealistically high stresses close to mounting locations. We propose to use "softened" Dirichlet conditions instead. To this end, we interpolate undeformed and deformed nodes on boundary triangles

$$\mathbf{X}(\boldsymbol{\zeta}) = \sum_{i=1}^{3} \mathbf{X}_i N_i(\boldsymbol{\zeta}) \quad \text{and} \quad \mathbf{x}(\boldsymbol{\zeta}) = \sum_{i=1}^{3} \mathbf{x}_i N_i(\boldsymbol{\zeta}) + \sum_{i=1}^{3} \hat{\mathbf{x}}_i \hat{N}_i(\boldsymbol{\zeta})$$

with shape functions of their respective elements. We then measure the distance between corresponding points on the undeformed and deformed contact surface, decoupling their difference in normal $\mathbf{n}(\zeta)$ and tangential directions (sliding)

$$d(\boldsymbol{\zeta}) = \frac{1}{2} (\mathbf{x} - \mathbf{X})^T \, \mathbf{D} \, (\mathbf{x} - \mathbf{X}) \quad \text{with} \quad \mathbf{D}(\boldsymbol{\zeta}) = w_\perp \mathbf{n}\mathbf{n}^T + w_\parallel (\mathbf{I} - \mathbf{n}\mathbf{n}^T).$$

Note that our distance measure resembles the potential energy of a linear spring ($\frac{1}{2} \times$ stiffness $\times$ squared displacement). Hence, we can interpret d as an energy density where the weights $w_\perp$ and $w_\parallel$ control the stiffness in normal and tangential directions. When setting weights, we use the stiffness of the shim material as reference.

Analogously to integration over volumetric elements, we split cut triangles into sub-triangles $\mathcal{T}_s$, and integrate the density d over the surface that is in contact with the mounting system

$$W_t = \sum_s \int_{\mathcal{T}_s} d(\boldsymbol{\zeta}) \det \left[\frac{\partial \mathbf{X}(\boldsymbol{\zeta})}{\partial \boldsymbol{\zeta}} \right] d\boldsymbol{\zeta}. \tag{7}$$

However, in contrast to our volumetric integration, we skip triangles (or sub-triangles) of elements (sub-elements) that are void.

For simulation, we add the additional degrees of freedom $\hat{\mathbf{x}}_i$ to our $3n$-vector $\mathbf{x}$, and set the internal energy W to $\sum_e W_e + \sum_t W_t$.

5 WORST-CASE LOADS

When optimizing the strength-to-weight ratio of structures, load cases have a significant influence on the performance of the result. While an optimized structure is stable under specified loads, perturbing the loads by small amounts can lead to failure. Analyzing several optimized structures, we observe that this is the rule rather than the exception. With our worst-case formulation, we aim at addressing this challenge, identifying the loads that lead to the weakest structure.

In the most general case, the pressure or traction distribution on a structure's entire surface is considered unknown. Bounding the maximum pressure, we could optimize the structure to withstand general load cases. However, this scenario is overly conservative and a large subset of loads are very unlikely or not even possible to reproduce in a practical setting. At smaller scales, an end-user might interact with printed parts in unpredictable ways [Ulu et al. 2017; Zhou et al. 2013], pressing and pulling on them. However, at larger scales, load cases are more clearly defined, with uncertainties spanning a small subspace.

Load types for large-scale structures are commonly specified by a structural engineer and include wind and live loads, emulating interactions if a structure is exposed to harsh weather conditions (e.g., a gargoyle on a cathedral) or is climbed onto or sat on. While we might estimate the maximum expected wind magnitude, the direction of the wind is largely unknown. Similarly, a structural engineer can advise in which regions a person could stand, hold

onto, or sit on a structure and what maximum weight we can expect. However, the specific locations within specified regions are again unknown.

We will first discuss loads without uncertainties, namely self-weight and thermal loads. While our worst-case optimization interfaces with arbitrary, user-defined loads, we will use wind loads to illustrate how uncertainty in directions is modeled. Live loads will serve as an example for loads with location uncertainties.

5.1 Specifying Load Cases

Due to the build material being weak and heavy, self-weight or so-called dead load is an important load case. In regions where the outside temperature varies significantly, thermal loads are induced: especially in proximity of mounting locations, very high or very low temperature can lead to high stresses. Thermal loads also further motivate our "softening" of Dirichlet conditions, since hard Dirichlet conditions would create unnaturally high stresses at mounting locations.

Self-Weight. Self-weight is a volumetric load caused by acceleration due to gravity in direction of the ground. Not making any assumptions about the orientation of our input, we let the user specify the 3D direction vector $\mathbf{g}$, setting its magnitude to the gravitational constant g. We then define an energy density

$$G(\boldsymbol{\xi}) = \rho(\boldsymbol{\xi})\mathbf{g}^T\left[\mathbf{x}(\boldsymbol{\xi}) - \mathbf{X}(\boldsymbol{\xi})\right]$$

that represents the work done by gravity (mass $\times\, g \times$ displacement) per unit volume, displacing a node within the structure.

Analogously to internal energy, we integrate over (sub-)elements

$$G_e = \sum_s \int_{\mathcal{E}_s} G(\boldsymbol{\xi}) \det\left[\frac{\partial \mathbf{X}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right] d\boldsymbol{\xi},$$

setting the density ρ at integration points in Ω_s to the density of the build material. Elements or sub-elements in the void space are skipped. To account for self-weight, we add $\sum_e G_e$ to our internal energy W.

We can also extend this potential to other types of volumetric loads. A common strategy to analyze the performance of structures under seismic loads is to apply loads in vertical and lateral directions, with magnitudes set to a fraction of the structure's weight. While we have not considered seismic loads when optimizing our demonstrations, we could use this quasi-static approximation to formulate a worst-case optimization, parameterizing the direction and magnitude of $\mathbf{g}$.

Thermal Loads. When a structure undergoes a change in temperature ΔT, its volume changes proportionally. Assuming our build material to be homogeneous and the temperature not to vary spatially, the volume change due to thermal expansion and contraction does not lead to internal stresses as long as the boundary is not fixed. This observation motivates the use of an offset deformation gradient

$$\hat{\mathbf{F}}(\boldsymbol{\xi}) = \mathbf{F}(\boldsymbol{\xi}) - \alpha\Delta T\mathbf{I}$$

where α is the *coefficient of thermal expansion* of the build material. Subtracting this offset changes the rest configuration of the structure, causing stresses to peak close to boundary conditions.

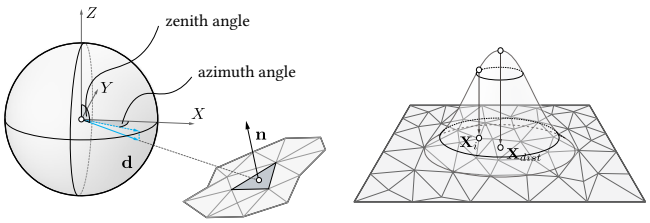

Fig. 6. **Parameterizing Wind and Live Loads** For wind loads, we parameterize the uncertainty in directions with spherical coordinates (left). For live loads, we parameterize the location of a load distribution with center $\mathbf{X}_{\text{dist}}$ (right).

5.2 Specifying Worst-Case Loads

For wind and live loads, there are uncertainties in the load's direction or location. Serving as illustrative examples, we will formalize the subspaces spanning these uncertainties with a set of parameters.

Wind Loads. Architectural ornaments are commonly subject to wind loads with a known magnitude but unknown direction. To parameterize unit directions, we rely on spherical coordinates, collecting the azimuth angle (in xy-plane from x-axis) and zenith angle (from z-axis) in a parameter vector. To avoid singularities at the two poles, we restrict the zenith angle to the range $[\varepsilon, \pi - \varepsilon]$. The azimuth angle can vary between 0 and 2π.

We apply wind loads to all or a user-selected subset of triangles. From the parameterized direction and the expected pressure of wind, we compute the load acting on an individual triangle by multiplying its projected area (area $\times$ cosine of angle between normal and wind direction) with the constant pressure (see Fig. 6 left). Multiplied with the unit wind direction, we add a third of the load to each incident node, repeating this procedure for all selected triangles.

In its simplest form, a wind load $\mathbf{l}_{\text{wind}}$ has one global direction and is acting on all triangles that face the direction. By combining the parameters of several directions in a vector $\mathbf{p}_{\text{wind}}$, we can model wind that acts on several disjoint or overlapping regions simultaneously.

Live Loads. When modeling live loads, we assume a load's overall magnitude, its direction, and its distribution kernel to be known, restricting uncertainty to locations of their distribution centers (compare with Fig. 6 right): a user first selects a subset of connected triangles on the structure at rest, then specifies an initial location $\mathbf{X}_{\text{dist}}$ within this region. The kernel function $w(\mathbf{X}, \mathbf{X}_{\text{dist}})$ is then used to compute weights for nodes that are a distance $\|\mathbf{X} - \mathbf{X}_{\text{dist}}\|$ away from the current center. To compute loads acting on selected triangle nodes $\mathbf{X}_i$, we distribute the constant 3D load $\mathbf{l}_{\text{dist}}$

$$\frac{w(\mathbf{X}_i, \mathbf{X}_{\text{dist}})}{\sum_j w(\mathbf{X}_j, \mathbf{X}_{\text{dist}})}\mathbf{l}_{\text{dist}},$$

normalizing the weights to ensure that the magnitude of the overall load equals the norm of the specified load $\mathbf{l}_{\text{dist}}$.

Assuming the curvature of the selected region to be small, we use a regularizer

$$\left\|\mathbf{X}_{\text{dist}} - \frac{\sum_i w(\mathbf{X}_i, \mathbf{X}_{\text{dist}})\mathbf{X}_i}{\sum_j w(\mathbf{X}_j, \mathbf{X}_{\text{dist}})}\right\|^2$$

to keep $\mathbf{X}_{\text{dist}}$ within the specified region. Note that only nodes of selected triangles contribute to the weighted average. Hence, we can interpret our regularizer as a spring that becomes active if the distribution center is moving away from the selected region. While small deviations of $\mathbf{X}_{\text{dist}}$ from the surface are possible, computed loads always apply directly on the surface.

Collecting several centers $\mathbf{X}_{\text{dist}}$ in a parameter vector $\mathbf{p}_{\text{live}}$, we can combine several distributions in a single load case $\mathbf{l}_{\text{live}}$, summing up their individual regularization terms in a regularizer $R(\mathbf{p}_{\text{live}})$. Like point loads in the discrete setting, our live loads can be understood as building blocks for more complex load cases in the continuous setting. Because regions a person can stand on, hold onto, or sit on tend to be flat, they approximate a large set of practical scenarios. Distribution kernels can either be measured or specified by the user, and we can use their support size to distribute an overall load to several locations. For our demonstrations, we use Gaussian kernels.

5.3 Optimizing for the Worst-Case

Given a specific parameterized load $\mathbf{l}(\mathbf{p})$, we aim at identifying the parameters $\mathbf{p}$ that maximize our failure potential f_{fail} under equilibrium constraints

$$\min_{\mathbf{p}} -f_{\text{fail}}(\mathbf{x}(\mathbf{p})) + R(\mathbf{p}) \quad \text{s.t.} \quad g_{\text{sim}}(\mathbf{l}(\mathbf{p}), \mathbf{x}(\mathbf{p})) = 0,$$

adding a regularization term where necessary. Note that changes to parameters change the load case, and the load case in turn changes the quasi-static equilibrium. We simplify this dependency $\mathbf{x}(\mathbf{l}(\mathbf{p}))$ in the problem above and the gradient derivations below.

For numerical optimization, we treat the equilibrium constraints implicitly, computing analytical gradients

$$g_{\text{fail}}(\mathbf{p}) = -\frac{\partial f_{\text{fail}}(\mathbf{x})}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}(\mathbf{p})}{\mathrm{d}\mathbf{p}} + \frac{\partial R(\mathbf{p})}{\partial \mathbf{p}}$$

of our combined objective $-f_{\text{fail}} + R$ with the help of the implicit function theorem

$$\frac{\mathrm{d}\mathbf{x}(\mathbf{p})}{\mathrm{d}\mathbf{p}} = -H_{\text{sim}}^{-1}(\mathbf{x}) \frac{\partial g_{\text{sim}}(\mathbf{l}, \mathbf{x})}{\partial \mathbf{l}} \frac{\partial \mathbf{l}(\mathbf{p})}{\partial \mathbf{p}}$$

where H_{sim} is the Hessian of our simulation objective.

As we describe in more detail in our supplemental material, we use the adjoint method for efficient computations. To handle additional constraints $\mathbf{c}(\mathbf{p})$ such as, e.g., the bounds on our wind load parameters, we use a Lagrangian $\mathcal{L}_{\text{fail}}(\mathbf{p}, \boldsymbol{\lambda}) = -f_{\text{fail}}(\mathbf{p}) + R(\mathbf{p}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{p})$ with multipliers $\boldsymbol{\lambda}$. Omitting an explicit treatment in the interest of clarity, we point the reader to related work [Xu et al. 2018] where first-optimality constraints on a Lagrangian are implicitly enforced. While our worst-case loads are parameterized with only a few parameters, our objective tends to have several local minima. To alleviate the problem of getting stuck in local minima, we sample in parameter space (we use 10 samples in our optimizations), setting the initial guess to the sample with the smallest objective value.

6 DESIGN OPTIMIZATION

In the last two sections, we described our simulation and introduced our worst-case load estimation. In this section, we will discuss how we evolve our discretized level set function to optimize the strength-to-weight ratio of an input structure.

Whenever we adjust the level set function $\boldsymbol{\phi}$, the rest configuration of a structure changes, and hence also its response $\mathbf{x}$. While worst-case loads tend to move if we change $\mathbf{X}$, the deformed configuration will changes even if the worst-case load stays the same. Captured in a nested continuous dependence $\mathbf{x}(\boldsymbol{\phi}, \mathbf{l}(\mathbf{p}(\boldsymbol{\phi})))$, we seek to optimize $\boldsymbol{\phi}$, measuring the structure's performance under resulting deformations $\mathbf{x}$.

To this end, we *nest* our simulation, worst-case load estimation, and design optimization, *unifying* the three problems into the *single continuous* optimization as outlined in Sec. 3

$$\begin{aligned} &\min_{\boldsymbol{\phi}} f_{\text{design}}(\boldsymbol{\phi}, \mathbf{x}(\boldsymbol{\phi})) \\ &\text{s.t.} \quad g_{\text{fail}}(\boldsymbol{\phi}, \mathbf{p}(\boldsymbol{\phi}), \mathbf{x}(\boldsymbol{\phi})) = 0 \\ &\qquad g_{\text{sim}}(\boldsymbol{\phi}, \mathbf{l}(\mathbf{p}(\boldsymbol{\phi})), \mathbf{x}(\boldsymbol{\phi})) = 0, \end{aligned}$$

simplifying the nested dependency in the interest of clarity.

6.1 Computing Gradients

Whenever we evaluate our design objective or its gradient, we minimize our worst-case load estimation to first-order optimality. Due to the static equilibrium constraint, first-order optimality of our load estimation also implies first-order optimality of our simulation objective.

Hence, we can compute analytical gradients of our design objective

$$g_{\text{design}}(\boldsymbol{\phi}) = \frac{\partial f_{\text{design}}(\boldsymbol{\phi}, \mathbf{x})}{\partial \boldsymbol{\phi}} + \frac{\partial f_{\text{design}}(\boldsymbol{\phi}, \mathbf{x})}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}(\boldsymbol{\phi})}{\mathrm{d}\boldsymbol{\phi}}$$

by applying the implicit function theorem to the first-order optimality constraints of our *unified* problem

$$\begin{bmatrix} \frac{\mathrm{d}\mathbf{p}(\boldsymbol{\phi})}{\mathrm{d}\boldsymbol{\phi}} \\ \frac{\mathrm{d}\mathbf{x}(\boldsymbol{\phi})}{\mathrm{d}\boldsymbol{\phi}} \end{bmatrix} = - \begin{bmatrix} H_{\text{fail}}(\boldsymbol{\phi}, \mathbf{p}, \mathbf{x}) & \frac{g_{\text{fail}}(\boldsymbol{\phi}, \mathbf{p}, \mathbf{x})}{\partial \mathbf{x}} \\ \frac{\partial g_{\text{sim}}(\boldsymbol{\phi}, \mathbf{l}, \mathbf{x})}{\partial \mathbf{l}} \frac{\partial \mathbf{l}(\mathbf{p})}{\partial \mathbf{p}} & H_{\text{sim}}(\boldsymbol{\phi}, \mathbf{x}) \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial g_{\text{fail}}(\boldsymbol{\phi}, \mathbf{p}, \mathbf{x})}{\partial \boldsymbol{\phi}} \\ \frac{\partial g_{\text{sim}}(\boldsymbol{\phi}, \mathbf{l}, \mathbf{x})}{\partial \boldsymbol{\phi}} \end{bmatrix}$$

where g_{fail}, g_{sim}, H_{fail}, and H_{sim} are gradients and Hessians of our failure and simulation objectives. To reduce the computational complexity, we again make use of the adjoint method, pointing the reader to our supplemental material for a detailed description.

Note that most entries of the gradient $g_{\text{design}}(\boldsymbol{\phi})$ are zero due to the following two reasons:

(1) all our three objectives f_{fail}, f_{weight}, and f_{CoM} depend on the domain Ω_s, and whenever we adjust $\boldsymbol{\phi}$, Ω_s changes. To update nodes on the interface, the two rest nodes $\mathbf{X}_i, \mathbf{X}_j$ connecting a cut edge (ϕ_i and ϕ_j have different signs) are linearly interpolated [Noël et al. 2015]. Because Ω_s only depends on a level set value ϕ_i if at least one of the incident tetrahedra is cut, the partial derivatives of f_{design} with respect to ϕ_i are zero except for nodes i close to the interface.
(2) the deformed configuration $\mathbf{x}(\boldsymbol{\phi})$ only depends on a ϕ_i if node i is incident to an enriched element (see Sec. 4). Hence, most columns of the Jacobian of $\mathbf{x}(\boldsymbol{\phi})$ with respect to the level set values $\boldsymbol{\phi}$ are zero.

Due to the many zero entries, the direct use of $g_{\text{design}}(\boldsymbol{\phi})$ is problematic. If we, during a level set evolution, moved the interface across element boundaries, the signed distance property would no

longer hold, hence convergence impeded. To propagate the gradient from the interface to the volume Ω, we use a normal velocity extension [van Dijk et al. 2013].

6.2 Evolving the Level Set

To evolve the interface, we rely on a simple Euler scheme

$$\boldsymbol{\phi}_{t+1}(\alpha) = \boldsymbol{\phi}_t - \alpha \frac{g_{\text{design}}(\boldsymbol{\phi}_t)}{\max_i |g_{\text{design}}(\phi_{t,i})|} \beta \tag{8}$$

where we normalize the entries of the extended gradient to the range $[-\beta, \beta]$, dividing them by the entry of maximum absolute value, followed by a scaling with factor β (set to 0.1 in all our experiments). To find the optimal α, we perform a standard backtracking line search, starting from $\alpha = 1$. We repeat this evolution step until the gradient is sufficiently small, or the change in the level set negligible.

To guarantee that the interface does not cross the user-defined design domain, we impose an upper bound on the level set values (node i with negative ϕ_i are part of the solid) that are outside the design domain but inside Ω. After every update, we enforce these constraints by projecting the level set values to this bound, setting the gradient entries to zero for values on the bound.

Note that our advection does not preserve the signed distance field property. Hence, we periodically reinitialize $\boldsymbol{\phi}$ with a simple shortest-path heuristic. While this is sufficient in our context, more elaborate reinitialization schemes exist [Morgan and Waltz 2017].

6.3 Updating Worst-Case Loads

After each evolution step, we generate new seeds (10 parameter samples) for each worst-case load $\mathbf{l}^k$. We then evaluate our worst-case objective $-f_{\text{fail}} + R$ for each seed, running a full load estimation only for the one with smallest objective value. If the structure is (1) weaker under the resulting load than under any of the existing load-parameter pairs and (2) the set of parameters is sufficiently different from existing ones, we *add* this load-parameter pair and then optimize the structure under all pairs (applied sequentially, not simultaneously). During design optimizations or when estimating worst-case loads, we keep refining the parameters for each pair, starting from the previously optimal set. If two sets of parameters become too close, we only keep one of them. Our worst-case load update can be understood as a stochastic process [Langlois et al. 2016], increasing robustness w.r.t. local minima.

7 RESULTS

We optimized the strength-to-weight ratio of a total of four models (see Figs. 1, 7, 8, 9, 10 and the accompanying video), tailored for 3D printing with Voxeljet's binder jetting technology. We printed three out of the four structures (Elephant, Stool, Gargoyle), and use the fourth result to discuss scalability (Dragon). For validation and discussion of failure criteria, level set initialization, and mesh resolution, we use a bridge example.

Binder Jetting & Infiltration. We printed our models on a Voxeljet VX1000. The VX1000 has a build volume of $1.0 \times 0.6 \times 0.5$ m, a resolution of 600 dpi in the x and y direction, and a layer thickness of 300 μm along the z axis. Silica Sand is used together with Voxeljet's phenolic resin binder. The remaining sand can be removed and recycled as we illustrate in the accompanying video.

To increase the strength and durability of printed structures, we can infiltrate them with an epoxy resin. For infiltration, the structures are best placed in a plastic bag, with an inlet pipe reaching into a container filled with resin, and an output pipe connected to a pump. Alternatively, we can place a part into a resin-filled container, and, in turn, place the container into a vacuum chamber. We infiltrated our Gargoyle and Stool examples.

Material Testing. To characterize material and strength parameters, we perform standard mechanical testing on cylindrical specimens: in a first test, we compress a cylinder along its axis, estimating the Young's modulus E and Poisson's ratio ν from the measured transverse and longitudinal strain. To estimate the compressive strength σ_c, we divide the failure load by the cylinder's cross-sectional area. The same testing machine can be used to characterize the tensile strength. To this end, we compress cylinders transversally (split-cylinder test). Hence, they fail due to tensile stresses. We then calculate the tensile strength by dividing the measured failure load by πrh where r is the radius and h the height of the specimen. To characterize the biaxial compressive strength of the material, advanced mechanical testing is required. We follow standard practice, setting σ_b to $1.2\sigma_c$. The material tests show a small anisotropy in material strength (print-direction dependence). We assume an isotropic material in our optimizations, setting the strength values to conservative estimates (smallest, measured values for σ_c, σ_t, and σ_b).

In our optimizations, we use $E = 0.66$ GPa, $\nu = 0.15$, $\rho = 1265\ \frac{\text{kg}}{\text{m}^3}$, $\sigma_t = 0.8$ MPa, $\sigma_c = 5.2$ MPa, and $\sigma_b = 6.2$ MPa for the uninfiltrated material, and $E = 2.43$ GPa, $\nu = 0.15$, $\rho = 1426\ \frac{\text{kg}}{\text{m}^3}$, $\sigma_t = 2.0$ MPa, $\sigma_c = 16.3$ MPa, and $\sigma_b = 20.0$ MPa for the epoxy-infiltrated material. To ensure structural integrity, we multiply the strength parameters with an additional safety factor of 0.5 before running optimizations.

Elephant. With our *Elephant*, we illustrate applications in designing stably-standing sculptures that are structurally sound under their own weight. To balance the model on one foot, we optimize our design objective f_{design} with the optional center of mass objective f_{CoM}. Only self-weight is considered, and no post-printing resin-infiltration is applied. In Fig. 7, we show the optimized result and the printed statue (top). The projected centers of mass and visualizations of the failure potential $s(\boldsymbol{\sigma})$ for the solid, the optimized, and the constant thickness models are shown at the bottom. Note that the constant thickness model uses the same volume as our optimized version, illustrating what an artist might do without the help of a structural optimization. As the projected center of mass is outside the support polygon for our solid and constant thickness models, they would both topple over under self-weight. The solid model would further *break under self-weight*, illustrating how weak the build material is and further motivating the need for a structural optimization.

Stool. Furniture, especially chairs and stools, have to withstand people sitting on them in various ways. For our *Stool*, we specify a live load with $\mathbf{l}_{\text{dist}} = 900$N (Gaussian distribution with spread 0.1 m), acting on the top surface of the model. The base is held fixed.

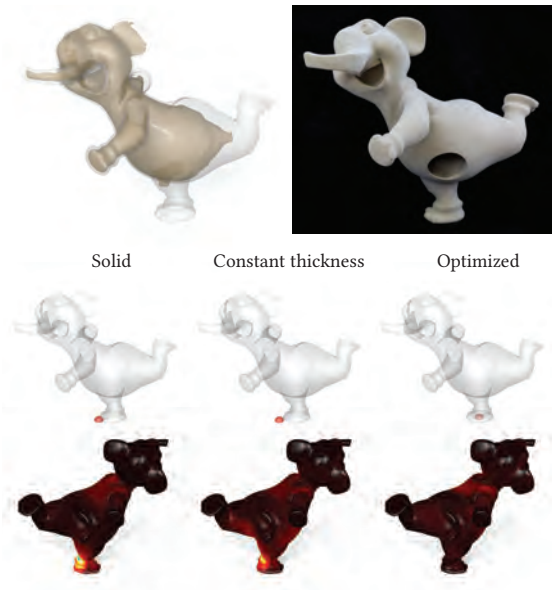

Fig. 7. Our *Elephant* is optimized to balance under self-weight (top). For the solid model (bottom left), the projected center of mass lies outside the support polygon. The visualization of the failure potential (bottom) shows that the model would break if fixed to the ground (blue region). The constant thickness model (bottom center), while structurally sound, would topple over. Our optimized model (bottom right) is sound *and* stands stably (see also image of printed model).

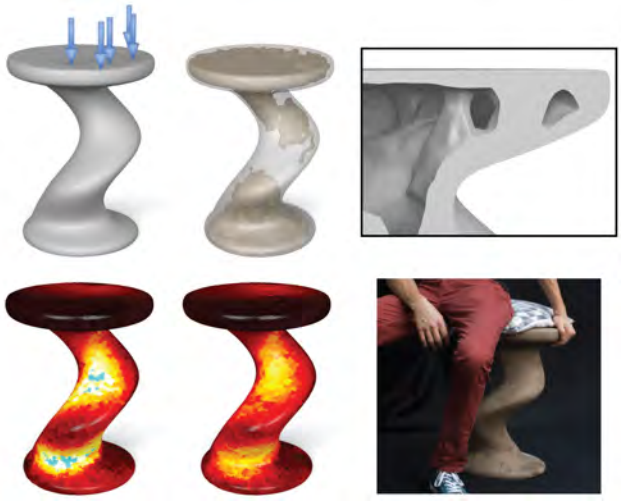

Fig. 8. For our *Stool*, we run a worst-case optimization under a live load that acts on the top surface, representing loads of a person sitting on the stool at unknown locations (bottom, right). The optimization identifies the worst-case loads (top left) that create the largest moments, reinforcing the center of the stool (top, center) and adding small struts to support the seat (top, right). Under worst-case loads, the optimized model is stable (bottom center), while a structure with the same volume but constant thickness would break (bottom left, areas with $s(\boldsymbol{\sigma})$ above 1 are marked in blue).

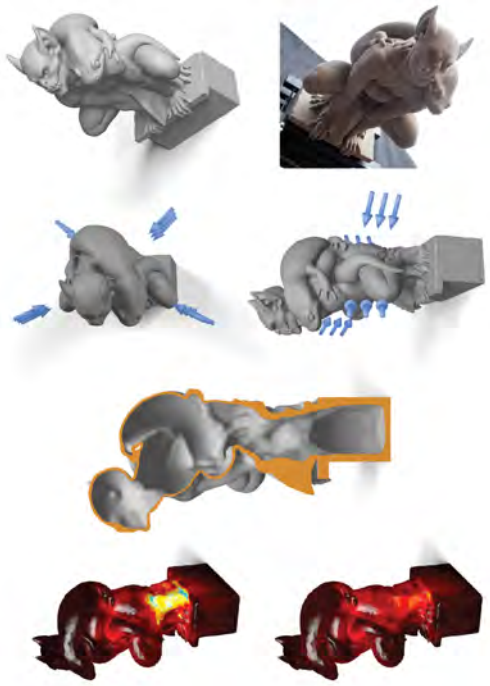

Fig. 9. The *Gargoyle* model (top left) is constrained at its base and subjected to four separate worst-case wind loads (second row). The optimization adds material to the connection between the gargoyle and the base (third row), where the wind induces the strongest moment. Looking at a specific wind load, the resulting structure is stable (bottom right), while a structure with the same volume, but constant wall thickness would break (bottom left, areas with a distance-to-failure $s(\boldsymbol{\sigma})$ above 1 are marked in blue).

During our worst-case optimization, a set of five worst-case loads are identified, all close to the boundary of the seat, hence inducing moments that lead to peak tensile stresses. The optimization adds small struts to distribute the loads (see Fig. 8 top right).

Gargoyle. As a typical decorative element on old cathedrals, gargoyles are prominent building features that experience large moments due to their own weight and strong winds. We apply our method to a *Gargoyle* model and, besides gravity, also consider worst-case wind loads with a wind pressure of 2.5 kPa. Fig. 9 shows the four worst-case wind directions identified during the optimization: two wind loads that increase the load in direction of gravity, and two wind directions that push the statue to the side, targeting weak spots at the feet. Our optimization successfully strengthens the model against these loads by thickening the parts of the base and its connection to the figure where tensile stresses tend to peak.

Asian Dragon. We demonstrate the scalability of our method on a virtual example, using the *Asian Dragon* model, scaled up to fill the build volume ($4 \times 2 \times 1\,m$) of the Voxeljet VX4000. Envisioning such a statue to be placed outside, we have to guarantee its stability under self-weight and temperature changes (+50°C and −30°C). In addition, we assume moderate live loads to act on the structure (e.g., a child sitting on the *Dragon*). To this end, we specify a worst-case

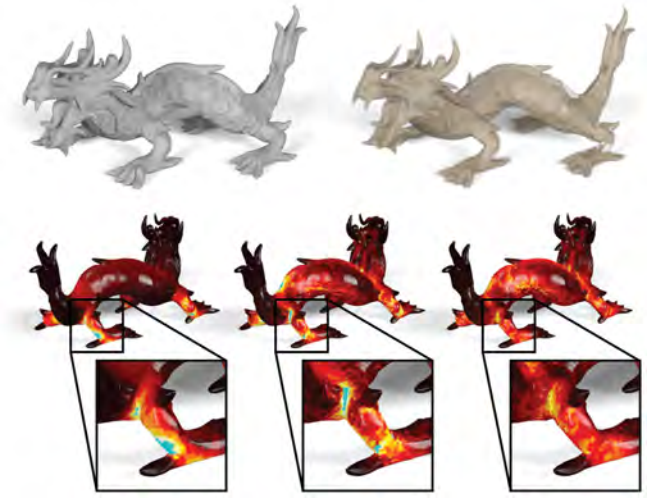

Fig. 10. We subject the *Asian Dragon* model (top left) to thermal and worst-case live loads, and compute the optimized internal geometry (top right). We show the plots of the distance-to-failure $s(\sigma)$ for the most extreme load case, thermal expansion, with values above 1 marked in blue (bottom). For this load case, the full model (bottom right) would break at the feet. The model with constant thickness and volume equal to the optimized model (bottom center) has less stress placed on its feet, but additional failure points along its back due to the thermal expansion. The optimized model (bottom right) alleviates stresses at these critical points.

Table 2. We report number of elements (# elems), dimensions (dim), number of iterations (# it), complexity of a single iteration (it. time), weight of the optimized structure (weight), and the ratio between optimized and initial volume (ratio) for our Elephant, Stool, Gargoyle, and Asian Dragon (demo). The timings were collected on a 4.0 GHz Intel Core i7-6700K quad-core processor with 32 GB of RAM.

demo	dim [cm]	#elems	#it	it. time [s]	weight [kg]	ratio [%]
Elephant	$58 \times 51 \times 71$	133197	12	130	8.5	34.8
Stool	$46 \times 46 \times 60$	30540	21	200	24.1	56.8
Gargoyle	$85 \times 49 \times 39$	132252	17	1401	19.2	31.9
Dragon	$229 \times 134 \times 100$	123963	54	458	49.2	16.2

live load of 50 kg, restricting its location to the region at the top of the center of the arch. We keep the feet fixed to the ground. Simulations show that the solid and constant thickness models would fail under thermal expansion (see Fig. 10). The optimization adds material at the feet, reducing excessive stresses due to the thermal expansion in the rest of the model. The live load, on the other hand, does not lead to critical stresses in any of the models.

Optimization. For our Elephant, we set the objective weights w_{fail}, w_{weight}, w_{CoM} to $4 \cdot 10^{10}$, 10^3, 10^4, respectively. For all other demonstrations, we use $w_{\text{fail}} = 4 \cdot 10^9$ and $w_{\text{weight}} = 10^4$. We set the exponent γ in our failure objective to 8 for all demonstrations, and 4 for our validation example (bridge). To set the weights for our "softened" Dirichlet conditions $w_\perp$ and $w_\parallel$, we use the approximate thickness and Young's modulus of the shim material. For our *Elephant* and *Gargoyle*, we chose a hard rubber material ($E = 0.1$ GPa) with a thickness of 1 cm, setting both weights to $10^{10} \frac{\text{N}}{\text{m}^3}$. For the Stool, we use the same rubber material for $w_\perp$. However, to allow for sliding, we set the parallel weight $w_\parallel$ to the significantly lower value $10^4 \frac{\text{N}}{\text{m}^3}$. Our *Dragon* model expands significantly due to thermal loads. To reduce stresses at the mounting locations, we assume the shim material to be 5 cm thick polystyrene foam ($E = 0.005$ GPa), setting the weights to $10^8 \frac{\text{N}}{\text{m}^3}$. We report key statistics including the weight of the optimized structure, the achieved volume reduction, and the overall time it took to optimize them in Tab. 2.

7.1 Validation

In the following section, we validate our choice of failure criterion and our worst-case load optimization, illustrating the sensitivity of our results with respect to changes in resolution and initialization on an example.

von Mises
Strength: 858.6 N

Bresler-Pister
Strength: 1075.8 N

Fig. 11. **Failure Criteria** For the bridge example, we tested fabricated examples optimized with the von Mises and Bresler-Pister failure criterion. For the same amount of material, the Bresler-Pister-optimized structure shows a higher strength.

Failure Criteria. We compare the Bresler-Pister failure criterion to the widely used von Mises criterion by measuring the strength of a bridge optimized using either of these failure criteria. Fig. 11 shows the two results, optimized for a load applied to the top center of the model. Both bridge designs have equal volume (we adjusted the objective weights accordingly). Since the Bresler-Pister criterion better approximates the true failure criterion, it manages the material budget more effectively. The result uses less material in regions of compression—the connection between struts and the sides of the bridge–and more material in the center, where tensile stresses appear. We fabricated these examples and tested them in a compression testing machine, revealing that the structures break under loads of 858.6 N and 1075.8 N for the von Mises and Bresler-Pister criterion, respectively (see accompanying video).

Mesh Resolution. We investigate the effect of the resolution of the underlying simulation mesh by optimizing the bridge example with a centrally applied fixed load. Fig. 12 shows that the optimization creates qualitatively similar meshes for a range of resolutions, with lower resolutions showing some discretization artifacts. Only the coarsest mesh resolution leads to a significantly different result.

Initialization. Our structural optimization is sensitive to the initialization as we illustrate in Fig. 13: if we start from the minimal volume (top, outer shell with user-specified thickness), the optimization converges to a solution with significantly less volume (but

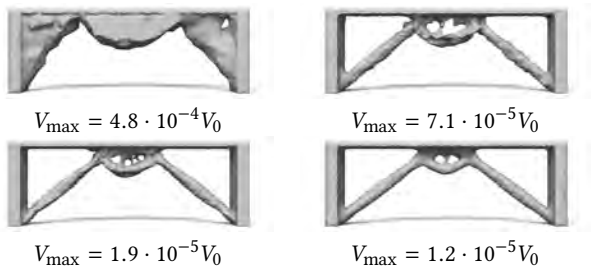

Fig. 12. **Mesh Resolution** We show the effect of changes to the resolution of the simulation on our bridge example. We report the maximum tetrahedron volume $V_{\max}$ for each of the results as a fraction of the total volume V_0. The optimization finds a strut-based solution even for relatively coarse meshes.

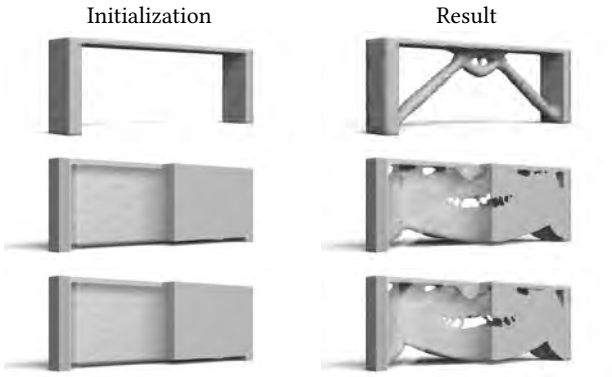

Fig. 13. **Initialization** We show the effect of the initialization (left) on the final result (right) of our optimization. Starting from a minimal volume, we arrive at a strut-based solution (top). With an initialization that adds material to the sides while leaving the center of the bridge void, we arrive at a different solution that uses more material (middle, cut open to reveal structure). Even an increase in resolution cannot push the optimization out of this local minimum (bottom).

similar strength) than if we start with material on the two sides of the bridge (two bottom rows). For the latter, the local minimum persists even if we increase the mesh resolution (bottom row). While our optimization will only reach a local minimum, our examples indicate that if we start from the minimal volume we can generally reach a solution that does not break under the specified loads, even though it might not be the globally optimal result that uses the least amount of material.

Failure Objective Energy Landscape. The behavior of the energy f_{fail} with respect to changes of a load's location illustrates how sensitive a model is to changes in load cases, motivating the use of worst-case loads. Fig. 14 shows the energy landscape of the bridge example, for a minimal volume model as well as models optimized under a fixed load and worst-case loads. We compute this energy landscape by varying the load location on top of the model, visualizing the position-dependent objective value. We observe that the energy is well-behaved, and our optimization finds the worst-case

Fig. 14. **Failure Objective Energy Landscape** We show the energy landscape (bottom) of the objective f_{fail} for all positions of a load applied to the top surface of the bridge. For an unoptimized bridge (left), the load location that leads to the highest energy—the location of the worst-case load—is in the center. For the fixed-load optimized bridge (center), the locations for the worst-case loads move to the two ends, and the effect of these loads is weaker. The bridge optimized under worst-case loads (right) no longer has any load locations that would lead to failure.

load positions in all cases. While the optimization under a fixed load leads to an intuitive solution with struts from the corners to the center, the model is susceptible to changes in load position. The model optimized under worst-case loads, on the other hand, shows no weak spots, having added more material to counteract all possible loads, favoring designs with a larger number of struts, or completely filled regions.

8 CONCLUSION

We presented a method for structural optimization that accounts for the asymmetric strength behavior of many common build materials, specifically considering binder-jetted sand prints, to create models with an optimized strength-to-weight ratio. We introduced a general framework for parameterized worst-case loads which models the inherent uncertainty of load cases, reducing the burden to find the optimal load parameters. We demonstrated its advantages on the continuous space of parameterized live and wind loads, where a manual design of an optimized model would be infeasible. Finally, we combined these components into a *unified continuous* optimization by nesting analysis, worst-case load estimation, and structural optimization, an approach with promising potential for applications beyond just structural optimization.

Limitations and Future Directions. Our worst-case loads only have very few parameters, and our add-but-not-replace worst-case optimization is effective in circumnavigating local minima. However, while we can significantly improve the strength-to-weight ratio of structures, we cannot guarantee global optimality.

The standard XFEM discretization has been known to overestimate the stress in elements where the level set cut results in a very small subelement. Since our stress objective is formulated as an integral, its sensitivity to these inaccuracies for higher stresses depends on the element size as well as the exponent γ: for high values of γ, we give inaccuracies in small subelements more weight while lower values lead to designs where material is used to strengthen regions with non-critical stresses. While these inaccuracies, for our choice of γ, remained within reasonable bounds for all our demonstrations, a specialized approach such as the robust stabilized stress computation [Sharma and Maute 2018] could improve the prediction accuracy and convergence of our method.

The Bresler-Pister criterion assumes isotropy. Mechanical tests showed that this assumption is valid for Voxeljet printers. Optimizing structures with anisotropies in strength and material parameters is, however, an exciting avenue of future work.

Even though build volumes will increase in size, there will be bounds on the size of models we can print. To make our method scalable and independent of the size of build trays, a decomposition into topologically interlocking building blocks [Fu et al. 2015; Molotnikov et al. 2015] provides an exciting direction for future research.

We do not model handling. While we have not observed any problems when printing and infiltrating our models, optimizing the structure for expected handling loads is an exciting future direction.

ACKNOWLEDGMENTS

The authors would like to thank Maurizio Nitti and Zhao Ma for model design; Espen Knoop, Zhao Ma, Hongyi Xu for help with video, figures, and mechanical testing; Jan Wezel, Mathias Bernhard, Coralie Ming, Ammar Mirjan, Andreas Reusser, and Pietro Odaglia for 3D printing and material characterization; Matthias Kohler and Fabio Gramazio for letting us use ETH's Voxeljet printer. The authors would also like to thank the anonymous reviewers for their valuable comments and helpful suggestions.

REFERENCES

Grégoire Allaire, Frédéric de Gournay, and François Jouve. 2009. Stress minimization and robust compliance optimization of structures by the level set method. In *Proc. 8th World Congr. Struct. Multidisciplinary Optimization.*

Grégoire Allaire, François Jouve, and Anca-Maria Toader. 2004. Structural optimization using sensitivity analysis and a level-set method. *Journal of computational physics* 194, 1 (2004), 363–393.

Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. 2014. Spin-it: Optimizing Moment of Inertia for Spinnable Objects. *ACM Trans. Graph.* 33, 4, Article 96 (July 2014), 10 pages.

Martin P. Bendsøe and Ole Sigmund. 2004. *Topology Optimization: Theory, Methods, and Applications.* Springer Berlin Heidelberg.

Amit H. Bermano, Thomas Funkhouser, and Szymon Rusinkiewicz. 2017. State of the Art in Methods and Representations for Fabrication-Aware Design. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 509–535.

Wai-Fah Chen and Atef F. Saleeb. 1994. *Constitutive Equations for Engineering Materials.* Elsevier.

Andrej Cherkaev and Elena Cherkaeva. 1999. *Optimal design for uncertain loading condition.* 193–213.

D-Shape. 2018. D-Shape. https://d-shape.com/. (2018). Accessed: 2018-06-04.

Frédéric de Gournay, Grégoire Allaire, and François Jouve. 2008. Shape and topology optimization of the robust compliance via the level set method. *ESAIM: Control, Optimisation and Calculus of Variations* 14, 1 (2008), 43–70.

Joshua D. Deaton and Ramana V. Grandhi. 2014. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization* 49, 1 (2014), 1–38.

Chi-Wing Fu, Peng Song, Xiaoqi Yan, Lee Wei Yang, Pradeep Kumar Jayaraman, and Daniel Cohen-Or. 2015. Computational Interlocking Furniture Assembly. *ACM Trans. Graph.* 34, 4, Article 91 (July 2015), 11 pages.

Timothy Langlois, Ariel Shamir, Daniel Dror, Wojciech Matusik, and David I. W. Levin. 2016. Stochastic Structural Analysis for Context-aware Design and Fabrication. *ACM Trans. Graph.* 35, 6, Article 226 (Nov. 2016), 13 pages.

Lin Lu, Andrei Sharf, Haisen Zhao, Yuan Wei, Qingnan Fan, Xuelin Chen, Yann Savoye, Changhe Tu, Daniel Cohen-Or, and Baoquan Chen. 2014. Build-to-last: Strength to Weight 3D Printed Objects. *ACM Trans. Graph.* 33, 4, Article 97 (2014).

Yangjun Luo and Zhan Kang. 2012. Topology optimization of continuum structures with Drucker-Prager yield stress constraints. *Computers & Structures* 90–91 (2012), 65–75.

Nicolas Moës, Mathieu Cloirec, Patrice Cartraud, and Jean-François Remacle. 2003. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering* 192, 28-30 (2003), 3163–3177.

Andrey Molotnikov, Ralf Gerbrand, Yuanshen Qi, George Philip Simon, and Yuri Estrin. 2015. Design of responsive materials using topologically interlocked elements. *Smart Materials and Structures* 24, 2 (2015), 025034.

Nathaniel R. Morgan and Jacob I. Waltz. 2017. 3D level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement. *J. Comput. Phys.* 336 (2017), 492–512.

Przemyslaw Musialski, Thomas Auzinger, Michael Birsak, Michael Wimmer, and Leif Kobbelt. 2015. Reduced-Order Shape Optimization Using Offset Surfaces. *ACM Trans. Graph.* 34, 4 (Aug. 2015), 102:1–102:9.

Lise Noël, Laurent Van Miegroet, and Pierre Duysinx. 2015. Analytical sensitivity analysis using the extended finite element method in shape optimization of bimaterial structures. *Internat. J. Numer. Methods Engrg.* 107, 8 (Dec. 2015).

Stanley Osher and Ronald Fedkiw. 2003. *Level Set Methods and Dynamic Implicit Surfaces.* Springer-Verlag New York.

Julian Panetta, Abtin Rahimian, and Denis Zorin. 2017. Worst-case Stress Relief for Microstructures. *ACM Trans. Graph.* 36, 4, Article 122 (2017), 16 pages.

Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. 2013. Make It Stand: Balancing Shapes for 3D Fabrication. *ACM Trans. Graph.* 32, 4, Article 81 (July 2013), 10 pages.

Ashesh Sharma and Kurt Maute. 2018. Stress-based topology optimization using spatial gradient stabilized XFEM. *Structural and Multidisciplinary Optimization* 57, 1 (Jan 2018), 17–38.

Jan Sokolowski and Jean-Paul Zolesio. 1992. *Introduction to Shape Optimization: Shape Sensitivity Analysis.* Springer-Verlag Berlin Heidelberg.

Ondrej Stava, Juraj Vanek, Bedrich Benes, Nathan Carr, and Radomír Měch. 2012. Stress Relief: Improving Structural Strength of 3D Printable Objects. *ACM Trans. Graph.* 31, 4, Article 48 (2012).

Erva Ulu, James Mccann, and Levent Burak Kara. 2017. Lightweight structure design under force location uncertainty. *ACM Trans. Graph.* 36, 4 (2017), 158.

Nobuyuki Umetani and Ryan Schmidt. 2013. Cross-sectional Structural Analysis for 3D Printing Optimization. In *SIGGRAPH Asia 2013 Technical Briefs.* Article 5.

Nico P. van Dijk, Kurt Maute, Matthijs Langelaar, and Fred Van Keulen. 2013. Level-set methods for structural topology optimization: a review. *Multidisciplinary Optimization* 48, 3 (2013), 437–472.

Laurent Van Miegroet and Pierre Duysinx. 2007. Stress concentration minimization of 2D filets using X-FEM and level set description. *Structural and Multidisciplinary Optimization* 33, 4 (Apr 2007), 425–438.

Voxeljet. 2018. Voxeljet. https://www.voxeljet.com/. (2018). Accessed: 2018-06-04.

Michael Yu Wang, Xiaoming Wang, and Dongming Guo. 2003. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering* 192, 1–2 (2003), 227–246.

Weiming Wang, Tuanfeng Y. Wang, Zhouwang Yang, Ligang Liu, Xin Tong, Weihua Tong, Jiansong Deng, Falai Chen, and Xiuping Liu. 2013. Cost-effective Printing of 3D Objects with Skin-frame Structures. *ACM Trans. Graph.* 32, 6, Article 177 (Nov. 2013), 10 pages.

Jun Wu, Niels Aage, Rüdiger Westermann, and Ole Sigmund. 2018. Infill Optimization for Additive Manufacturing—Approaching Bone-Like Porous Structures. *IEEE Transactions on Visualization and Computer Graphics* 24, 2 (Feb 2018), 1127–1140.

Jun Wu, Christian Dick, and Rudiger Westermann. 2016. A System for High-Resolution Topology Optimization. *IEEE Transactions on Visualization and Computer Graphics* 22, 3 (March 2016), 1195–1208.

Hongyi Xu, Espen Knoop, Stelian Coros, and Moritz Bächer. 2018. Bend-It: Design and Fabrication of Kinetic Wire Characters. *Proc. of ACM SIGGRAPH Asia* 37, 6 (2018).

Haiming Zhao, Weiwei Xu, Kun Zhou, Yin Yang, Xiaogang Jin, and Hongzhi Wu. 2017. Stress-Constrained Thickness Optimization for Shell Object Fabrication. *Computer Graphics Forum* 36, 6 (2017), 368–380.

Qingnan Zhou, Julian Panetta, and Denis Zorin. 2013. Worst-case Structural Analysis. *ACM Trans. Graph.* 32, 4, Article 137 (2013).

Yahan Zhou, Evangelos Kalogerakis, Rui Wang, and Ian R. Grosse. 2016. Direct shape optimization for strengthening 3D printable objects. *Computer Graphics Forum* 35, 7 (2016).

APPENDIX

The constant coefficients of the Bresler-Pister criterion are parameterized with the compressive σ_c, the tensile strength σ_t, and the biaxial compressive strength σ_b of the build material:

$$A(\sigma_t, \sigma_c, \sigma_b) = \frac{1}{\sqrt{3}} \frac{\sigma_c \sigma_b \sigma_t (\sigma_t + 8\sigma_b - 3\sigma_c)}{(\sigma_c + \sigma_t)(2\sigma_b - \sigma_c)(2\sigma_b + \sigma_t)}$$

$$B(\sigma_t, \sigma_c, \sigma_b) = \frac{1}{\sqrt{3}} \frac{(\sigma_c - \sigma_t)(\sigma_b \sigma_c + \sigma_b \sigma_t - \sigma_c \sigma_t - 4\sigma_b^2}{(\sigma_c + \sigma_t)(2\sigma_b - \sigma_c)(2\sigma_b + \sigma_t)}$$

$$C(\sigma_t, \sigma_c, \sigma_b) = \frac{1}{\sqrt{3}} \frac{3\sigma_b \sigma_t - \sigma_b \sigma_c - 2\sigma_c \sigma_t}{(\sigma_c + \sigma_t)(2\sigma_b - \sigma_c)(2\sigma_b + \sigma_t)}$$