Fig. 1. Our pipeline optimizes for a geometry that achieves a target release curve. Left: starting from the density field at the current iteration, we simulate the release process by computing the dissolution time as the solution to a variable-speed Eikonal equation and evaluating the release curve by calculating an integral over the domain. The deviation from the target release profile is then measured and differentiated with respect to the density variables to obtain the design objective gradient and update the design. Right: visualization of the volumetric density fields describing the evolving 3D design along with the final extracted object and fabricated result.

Objects with different shapes can dissolve in significantly different ways inside a solution. Predicting different shapes’ dissolution dynamics is an important problem especially in pharmaceutics. More important and challenging, however, is controlling the dissolution via shape, i.e., designing shapes that lead to a desired release behavior of materials in a solvent over a specific time. Here, we tackle this challenge by introducing a computational inverse design pipeline. We begin by introducing a simple, physically-inspired differentiable forward model of dissolution. We then formulate our inverse design as a PDE-constrained topology optimization that has access to analytical derivatives obtained via sensitivity analysis. Furthermore, we incorporate fabricability terms in the optimization objective that enable physically realizing our designs. We thoroughly analyze our approach on a diverse set of examples via both simulation and fabrication.

1 INTRODUCTION

The dissolution process is a ubiquitous yet fascinating phenomenon. Objects with the same mass but different shapes can dissolve via different dynamics, resulting in vastly different release profiles. This property can be exploited, particularly in pharmaceutics, when designing controlled-release systems for drugs, food, and cosmetics. The surge of 3D printing with its unique capabilities for fabricating complex shapes opens new doors for devising novel controlled-release designs [Goole and Amighi 2016].

The first step toward design of controlled-release structures is to understand and model the dissolution process as a function of the shape. Such forward models of dissolution have been developed at different levels of abstraction from submolecular simulations [Sagui...
and Darden 1999] to more approximate but efficient Monte Carlo approaches [Briese et al. 2017] to continuum mechanics models [Zunino et al. 2009]. However, finding a design that matches a desired release profile using only a forward model would require a painstaking manual trial-and-error approach. An outright inverse design strategy, where the design is directly optimized to match a target release profile, is crucial to efficiently discover performant designs. Although a handful of existing works address the problem of tuning release profiles [Lu and Anseth 1999], they are limited to predefined setups, such as multi-laminated systems and handicap assumptions, such as one-dimensional dissolution. To the best of our knowledge, we are the first to consider the problem of shape from release for freeform 3D printable geometries.

We propose a gradient-based inverse design framework for the problem of architected release. The backbone of our solution is a simple forward model, based on a geometric abstraction, that is governed by the Eikonal equation. We formulate our problem as a PDE-constrained topology optimization whose objective is to fit the release curve computed from the Eikonal equation solution to the target release profile. We present an exact sensitivity analysis of the discrete fast marching solution and obtain a robust and efficient simulation model with analytical derivatives. Our formulation enables gradient-based optimization for discovering shapes achieving novel release behaviors and exploring trade-offs between release curve closeness and fabricability. We thoroughly analyze our approach on a diverse set of release profiles, both hand crafted and from known shapes. In summary, our contributions are:

- A simple and efficient geometric approach to the release simulation problem that is competitive in accuracy with past methods while being differentiable.
- An adjoint sensitivity analysis for our forward model to support fast gradient calculation.
- A topology optimization algorithm for inverse design of 3D printable geometries that match a prescribed release curve.
- A novel image-based measurement setup for acquiring the physical release curves of 3D printed objects.

2 RELATED WORK

Inverse Design for Functional Fabrication. Functional fabrication refers to generating designs that follow functional goals, or performances [Bermano et al. 2017]. The computational design and fabrication community has invented many inverse design frameworks to achieve a diverse set of functional goals related to mechanical properties [Schumacher et al. 2015; Stava et al. 2012; Bickel et al. 2010; Prévost et al. 2013], appearance [Sumin et al. 2019; Babaei et al. 2017; Matusik et al. 2009; Mitra and Pauly 2009; Schüller et al. 2014; Schwartburg et al. 2014; Fioravanti et al. 2020], motion [Bächer et al. 2014; Tang et al. 2020; Coros et al. 2013], photonics [Auzinger et al. 2018; Minkov et al. 2020], buoyancy [Wang and Whiting 2016], acoustics [Li et al. 2016], and more. Most computational approaches solve functional fabrication problems by first simulating the forward process and then inverting the simulation in order to estimate designs meeting a given target performance. In recent years, additive manufacturing has raised new opportunities and challenges for computational design. The most important challenge is navigating the vast design space, leading to the increasing popularity of topology optimization (TO). Once primarily concerned with structural performance [Bendsoe and Sigmund 2013], TO is now the go-to tool for many computationally complex design problems spanning a broad range of physics and application domains [Osanov and Guest 2016]; it enables the discovery of optimal geometries entirely from scratch, requiring no application-specific design parametrization, prior knowledge of the design’s topology, or expert-provided initial guess. Our work demonstrates that, thanks to our geometric abstraction of the dissolution process, the functional fabrication problem of inverse release curve design can be tackled by an efficient TO algorithm.

Additive Manufacturing of Release Structures. 3D printing of devices (e.g., tablets) with controlled release has a relatively long history in pharmacy [Goole and Amighi 2016] provide a comprehensive review. Multi-material printing, mostly binder-jetting, offers considerable flexibility for designing release profiles as active ingredients can be encapsulated in placebo barrier scaffolds [Katstra et al. 2000; Lam et al. 2002; Thakral et al. 2013]. Filament Deposition Modelling (FDM) has recently gained significant popularity for 3D printing tablets with controlled release [Goyanes et al. 2015; Skowyra et al. 2015; Pietrzak et al. 2015]. So far, most studies of FDM printed structures have focused on empirical evaluation of the effect of the geometry and infill patterns on the release profile. In this work, we employ an FDM 3D printer as our fabrication platform and use a water-soluble filament, typically used as the support material, as the solute.

Forward and Inverse Release. The dissolution process, and other phenomena emerging from intermolecular interactions, are studied in thermodynamics and physical chemistry. These molecular (and submolecular) phenomena can be modeled from first principles via, for example, Density Functional Theory (DFT) [Dreizler and Gross 2012]. Although these first-principles simulations are powerful and general [Sagui and Darden 1999], they are limited at most to a few thousands of atoms. Therefore, higher-level methods like stochastic simulations are popular for capturing the dynamics of larger systems, including the dissolution process [Briese et al. 2017]. Nisser et al. [2019] use a Monte Carlo (MC) approach to predict the overall dissolution time of 3D printed water-soluble support structures. In their approach, each voxel is considered as an atom in a crystalline lattice structure and is eroded with some probability at each simulation time step depending on its connectivity to other voxels. In this work, we compare our proposed distance-based forward model to the MC approach. Siepmann and Siepmann [2008] provide a broader survey of the mathematical models of the release process that have been developed for various device types; most either neglect the effects of geometry or are derived for simple geometries like spheres and cylinders, for which analytical formulas can be obtained. Our method can be seen as a generalization of the semi-empirical formulas derived in [Hopfenberg 1976; Cooney and Gross 2012], reformulating the simulation as an Eikonal equation to address arbitrary geometries. To the best of our knowledge, the problem of computational inverse design of controlled release structures has only been considered for multi-laminates, i.e., matrix devices where strata with
different concentrations or thicknesses are juxtaposed [Georgiadis and Kostoglou 2001; Nauman et al. 2010; Zhang 2020; Lu and Anseth 1999]. The common goal of these works is to achieve a desired release profile by tuning the concentrations and/or thicknesses of a handful of layers. For computational simplicity, all but one face of the matrix are insulated using an impermeable shell resulting in a one-dimensional diffusion problem. The forward simulation is then modelled using Fick’s law [Fick 1855; Treybal 1980] and the inverse design is formulated as a continuous optimization which is solved via variational calculus [Lu et al. 1998]. Our computational approach works for single-material designs, thus removing the need for a second material, i.e., an impermeable placebo shell. Furthermore, compared to the multi-stage manufacturing of multilaminates, our method enjoys a practical advantage.

**Optimal Control of the Eikonal Equation.** Using our geometric abstraction, the controlled release design problem can be posed as a PDE-constrained optimization where the state equation (constraint) is the Eikonal equation. This same mathematical formulation has been used to solve travel-time tomography problems in seismology and electromagnetism [Leung and Qian 2006] and, on the design side, to control the propagation of light within a medium by optimizing its composition [Seyedrezaei and Dadashzadeh 2016]. These works use the iterative fast sweeping method to solve both the forward and adjoint equations, but [Deckelnick et al. 2011] show how both can be solved using the efficient fast marching method; the approach we take is essentially equivalent, but we provide a more accessible derivation (Section 3.5) and address fully general objective functions rather than the specific tomography objective. Finally, Ospald and Herzog [2017] use the Eikonal equation to estimate the fiber orientation produced by the injection molding process for fiber-reinforced plastics, whose anisotropic stiffness strongly influences the elastic behavior of the resulting structure; they propose a compliance minimization TO problem constrained by both linear elasticity and the Eikonal equation to account for this anisotropy during structural optimization.

## 3 METHOD

Our central goal is to formulate a differentiable forward model of the release process for a given object geometry $\Omega \subset \mathbb{R}^3$ and then employ it in a gradient-based design optimization framework. The following subsections detail the components of our full system for designing 3D printable geometries to achieve an input release curve.

### 3.1 Simulation of a Fully Solid Object

Motivated by the Noyes-Whitney equation (Appendix A), we propose a simplified geometric model for simulating the release of an object into a stirred solvent that is competitive with other more established simulation approaches and yet is amenable to efficient sensitivity analysis. The fundamental underlying assumption of our model is that the entire exposed boundary of the object dissolves at the same rate. This rate will depend on the current concentration level in the solvent, but to simplify the exposition, we further suppose that the rate is constant. We show in Appendix B that concentration-dependent rates can be handled by a nonlinear remapping of the simulation time, however this was not necessary in the concentration regime of our dissolution experiments.

By defining the simulation time unit to be the length of time required to dissolve through a thickness of 1 mm, we conclude that the time at which a given point $x$ within a fully solid design $\Omega$ dissolves is the point’s distance $u(x)$ to the boundary $\partial \Omega$ (Figure 2). Accordingly, the cumulative release at simulation time $\tau$ can be computed directly from the distance field $u : \Omega \rightarrow \mathbb{R}^+$:

$$ C(\tau) = \int_{\Omega} \rho_0 (u < \tau) \, dx = \int_{\Omega} \rho_0 H(\tau-u) \, dx, \quad H(t) \equiv \begin{cases} 1 & t > 0 \\ 0 & t \leq 0 \end{cases}, $$

where $\rho_0$ is the mass density of the fabrication material, and $H$ is the Heaviside step function. The distance field $u$ can be computed in a number of ways (e.g., fast marching [Sethian 1999] or the Geodesics in Heat approach [Crane et al. 2017]) and can be characterized as a viscosity solution of the unit-speed Eikonal equation:

$$ \|\nabla u\| = 1 \quad \text{in} \quad \Omega $$

$$ u = 0 \quad \text{on} \quad \partial \Omega. $$

### 3.2 Simulation of a Density Field

The formula for $C(\tau)$ just presented has a few limitations. It can simulate only fully solid objects without holes, and it is not clear how to incorporate it in a topology optimization. To address both limitations, we generalize our simulation model by adopting a density-based object representation. Denoting the rectangular design domain (e.g., a 3D printer’s build volume) by $D \subset \mathbb{R}^3$, we represent the object by a design density function $\rho : D \rightarrow [0, 1]$ (distinct from the physical material density $\rho_0$, which we assume in the following to be $\rho_0 = 1$ without loss of generality). For instance, to represent $\Omega \subseteq D$, now possibly with holes, we take $\rho$ to be the indicator function $\chi_{\Omega}$, which assumes value 1 inside $\Omega$ and 0 outside. Our insight for simulating an object of this type is to interpret low density values as material that dissolves at extremely high speed and solve a variable-speed Eikonal equation on $D$:

$$ \|\nabla u\| = n \quad \text{in} \quad D $$

$$ u = 0 \quad \text{on} \quad \partial D, $$

(1)
where \( n : \mathcal{D} \rightarrow (0, \infty) \) is the reciprocal dissolution speed field sometimes called the "slowness field" (speed is \( 1/n \)). This formulation automatically handles structures like interior holes and nested shells, where the inner shell starts dissolving immediately after the outer shell fully dissolves (Figure 2). We use an interpolation law to define \( n \) in terms of the design density:

\[
 n = \rho + (1 - \rho)n_{\text{min}},
\]

where \( n_{\text{min}} \) is a small value used to avoid numerical issues in the Eikonal equation solver associated with extreme speeds (we set \( n_{\text{min}} = 10^{-6} \) for all experiments). This simplest choice of a linear interpolation law is natural since one would expect, e.g., a half-occupied voxel to dissolve in half the time. Note that since we will penalize our final designs to be nearly binary, the precise choice of interpolation law only influences intermediate designs evaluated during the optimization.

After solving for the "dissolution time" field \( \tau \), we evaluate the cumulative release curve by integrating over the full design domain:

\[
 C(\tau) = \int_{\mathcal{D}} \rho \, H(\tau - u) \, dx.
\]

### 3.3 Topology Optimization Formulation

Given a target release curve \( C^*(\tau) \), we formulate the inverse design problem as the optimization:

\[
 \min_{0 \leq \beta \leq 1} J_{\text{fit}}(\beta) + w_b J_{\text{binary}}(\beta) + w_f J_{\text{fab}}(\beta)
\]

\[
 J_{\text{fit}} := \frac{1}{T} \int_0^T (C(\tau) - C^*(\tau))^2 \, dt
\]

\[
 J_{\text{binary}} := \int_{\mathcal{D}} \rho (1 - \rho) \, dx
\]

\[
 J_{\text{fab}} := \int_{\mathcal{D}} \rho \, w \, dx
\]

where \( \text{smooth}(\cdot, r) \) is a standard smoothing filter with a kernel that linearly decays to 0 over the smoothing radius \( r \) [Bruns and Tortorelli 2001], and \( \text{project} \) is the threshold projection filter from [Wang et al. 2011]:

\[
 \text{project}(\hat{\rho}, \beta) := \frac{\tanh(\beta/2) - \tanh(\beta(\hat{\rho} - 1/2))}{2 \tanh(\beta/2)}.
\]

Together, these filters avoid nonphysical dependency issues like checkerboard patterns while retaining a binary design. \( J_{\text{binary}} \) is a standard term penalizing non-binary designs [Schumacher et al. 2015], and \( J_{\text{fab}} \) defined in Section 3.6 is a term promoting robustly printable designs. The total simulation duration \( T \) is determined by the maximum possible dissolution time for a fully solid design; for a cube-shaped \( \mathcal{D} \), this is half the cube width.

### 3.4 Discretization and Solvers

We discretize the design domain \( \mathcal{D} \) with a regular grid of \( N \) cube voxels having edge length \( \Delta x \). The density \( \rho \) is discretized as a piecewise constant field, meaning we define one design variable per voxel. When solving (1), we pad the grid with a layer of zero-density voxels on which the \( u = 0 \) boundary condition is applied. This is to ensure that parts of the object that intersect the domain boundary are not considered already dissolved at \( \tau = 0 \).

![Fig. 3. Simulation process in 2D: using the density and fast marching dissolution time fields. (a) We sum each voxel’s linear ramp release curve (visualized in (b) for the three highlighted voxels) to obtain the discrete release curve (c). Under grid refinement, the resulting discrete approximation exhibits first-order convergence to the continuous release curve \( C(\tau) \), validated in (d) using an analytical ground-truth formula for the disk’s release.](image)

We solve (1) using the implementation of the fast marching method provided by scikit-fmm [Furtney 2021], which we modified to track the information necessary for performing the exact adjoint sensitivity analysis discussed in the next subsection. This method is based on a finite difference discretization of the Eikonal equation, and the code supports a second-order scheme for improved accuracy. Unfortunately, the more complicated stencil selection rules involved in the second-order scheme degrade the differentiability of solution \( u \) (and thus the computed release curve), and so we used the first-order scheme for all inverse design. Denoting the dissolution time at voxel \( i \) by \( u_i \) and the neighbor of voxel \( i \) in the positive or negative direction along the \( d \) axis as \( N(i, \pm d) \), the first-order scheme in 3D solves the discrete equation:

\[
 \sum_{d=1}^{3} \frac{1}{(\Delta x)^2} \max(u_i - u_{N(i,d)}, u_i - u_{N(i,-d)}, 0)^2 = n_i^2 \quad \forall i.
\]

Note that the max operation automatically selects the correct upwind stencil for each dimension.

The dissolution time \( u_i \) determined by (5) indicates the time at which the voxel \( i \) has fully dissolved; this interpretation is apparent from studying a 1D version of the discretization. It would be both nonphysical and yield a nondifferentiable model to assume that the entire voxel dissolves instantaneously at time \( u_i \). Instead, we assume that the voxel dissolves at a constant rate over the time interval \( (u_i - n_i \Delta x, u_i) \). In other words, voxel \( i \) contributes to the release curve a scaled linear ramp function:

\[
 \tilde{C}_i(\tau) := \rho_i (\Delta x)^3 \text{clamp}\left( \frac{\tau - u_i}{n_i \Delta x} + 1, 0, 1 \right).
\]

Summing these contributions over the voxels obtains the following discretization of (3):

\[
 \tilde{C}(\tau) := \sum_{i=1}^{N} \tilde{C}_i(\tau) = \sum_{i=1}^{N} \rho_i H\left( \frac{\tau - u_i}{n_i \Delta x} \right) (\Delta x)^3.
\]
where we recognize $\tilde{H}(\tau) := \text{clamp}(\tau + 1, 0, 1)$ to be a smeared version of the Heaviside function.

We discretize the time integral in $J_{\text{fit}}$ by picking a time step size $\Delta t$ and computing \( \frac{1}{T} \sum_i (C(i\Delta t) - C'(i\Delta t))^2 \Delta t \). We use $\Delta t = T/1000$ for our results. We note that a literal implementation of (7), summing up the full vector of concentration samples $\tilde{C}(i\Delta t)$, has computational complexity $O(N^2)$, which becomes a bottleneck when high temporal resolution is requested. This can be accelerated to $O(N + T/\Delta t)$ by exploiting the fact that the second derivative of ramp function (6) is highly sparse, consisting of just two delta functions; therefore, we can accumulate the second finite difference of the discrete release curve in $O(N)$ and then integrate it twice (with cumulative sums) to obtain $\tilde{C}(i\Delta t)$.

We solve the discrete version of (4) using Knitro’s L-BFGS implementation. This algorithm requires the gradients of the design objectives $J_{\text{fit}}, J_{\text{binary}}, J_{\text{fab}}$ with respect to the filtered density variables $\rho$ which are then back-propagated through the density filter chain to obtain the gradients with respect to the design variables $\hat{\rho}$. $J_{\text{binary}} = \sum_{i=1}^N \rho(i-1) \Delta x^3$ can be directly differentiated, and $J_{\text{fab}}$ will be defined and differentiated in Section 3.6. Differentiating $J_{\text{fit}}$ is more challenging since it requires derivatives of our forward model; we obtain these efficiently with exact sensitivity analysis using the adjoint method as described in the next section.

### 3.5 Sensitivity Analysis

While a continuous adjoint equation can be derived for (1), care must be taken to discretize it consistently with the discretization of (1) itself. We prefer to derive the consistent adjoint equation for the discrete equation directly. We begin with an abstract formulation of the discrete Eikonal equation (5):

$$c(u, n) = 0,$$  \hspace{1cm} (8)

where $u$ and $n$ are vectors collecting the dissolution times and reciprocal speeds of each voxel, and $c$ is a vector of nonlinear constraint functions, one per voxel. Interpreting this equation as implicitly defining a dissolution time function $u(n)$, we consider a generic objective function $\tilde{f}(n) = \hat{f}(u(n), n)$ and seek its gradient:

$$\frac{\partial \tilde{f}}{\partial n} = \frac{\partial \hat{f}}{\partial u} \frac{\partial u}{\partial n} + \frac{\partial \hat{f}}{\partial n}.$$  \hspace{1cm} (9)

where $\frac{\partial u}{\partial n}$ is the derivative of the solution to the discrete Eikonal equation that we obtain by differentiating both sides of (8):

$$\frac{\partial c}{\partial u} \frac{\partial u}{\partial n} = -\frac{\partial c}{\partial n} \quad \iff \quad \frac{\partial u}{\partial n} = -\left( \frac{\partial c}{\partial u} \right)^{-1} \frac{\partial c}{\partial n}.$$

Substituting this expression into (9) obtains an explicit gradient:

$$\frac{\partial \tilde{f}}{\partial n} = -\frac{\partial \hat{f}}{\partial u} \frac{\partial c}{\partial u}^{-1} \frac{\partial c}{\partial n} + \frac{\partial \hat{f}}{\partial n} = -\lambda \frac{\partial c}{\partial n} + \frac{\partial \hat{f}}{\partial n} = \lambda \text{diag}(2n) + \frac{\partial \hat{f}}{\partial n},$$

which is efficient to calculate after solving the adjoint equation:

$$\left( \frac{\partial c}{\partial u} \right)^T \lambda = \frac{\partial \hat{f}}{\partial n}.$$  \hspace{1cm} (10)

### 3.6 Fabrication Constraints

To simplify fabrication, we 3D print our geometries without support structures. Therefore they must be free of overhanging features. We do this in one of two ways: (1) generating 2D extruded designs or (2) using the layer-by-layer additive manufacturing approach proposed in [Allaire et al. 2017] that we have adapted to our setting of density-based topology optimization.

For our 2D extrusion design space, we define the density variables on a 2D grid that is replicated in the third dimension up to an extrusion height $h$, which the user can specify according to, e.g., packaging concerns. We perform the fast marching simulation and sensitivity analysis on the 3D grid and then backpropagate the 3D gradient through the replication step to get the gradient in the 2D design space.

For our fully 3D designs, we evaluate the structural compliance under self weight of a small number of partial designs created at intermediate steps of the incremental, layer-by-layer fabrication process. We refer to the average compliance across all simulations as the incremental compliance objective (ICO), which we use as $J_{\text{fab}}$. Specifically, we divide the design along the build direction into $L$ layers, and for the design consisting of layers $1$ through $L$, we perform a linear elasticity finite element simulation under self weight load $f^{(L)}(\rho)$. This entails solving the linear system

\[ ACM Trans. Graph., Vol. 41, No. 6, Article 274. Publication date: December 2022. \]
We fabricate all models using a consumer-level, FDM 3D printer, and we use a high-performance multigrid-preconditioned conjugate gradient solver. The stiffness matrix $K$ is determined from $\rho$ using the standard SMP interpolation law using the typical penalization parameter setting $p = 3$. We then evaluate the incremental compliance:

$$J_{\text{fab}} := \frac{1}{2L} \sum_{l=1}^{L} f^{(l)}(\rho) \cdot u^{(l)}.$$

Structural compliance is self-adjoint, meaning we can efficiently compute its gradient without solving an adjoint equation:

$$\frac{\partial J_{\text{fab}}}{\partial \rho} = \frac{1}{L} \sum_{l=1}^{L} \left( \frac{\partial f^{(l)}}{\partial \rho} \cdot u^{(l)} - \frac{1}{2} \frac{\partial u^{(l)}}{\partial \rho} \cdot \partial K^{(l)} \cdot u^{(l)} \right).$$

4 EXPERIMENTAL SETUP

We validate our simulation model and inverse design pipeline by 3D printing geometries and measuring their release using a custom experimental setup detailed in the following subsections.

4.1 Fabrication

We fabricate all models using a consumer-level, FDM 3D printer (Prusa i3 MK3s) equipped with a 0.2 mm Kaika nozzle from Tecdia Co. We use AquaSys® 120 from Infinite Materials Solutions™, a water-soluble filament typically used as support structure. We tune the printing profile (CURA 4.13 slicer) to achieve a consistent average print density across different objects (1.32 g/cm³) and set the nozzle and bed temperatures to the manufacturer recommendations of 245 °C and 85 °C, respectively. The line width is set to 0.2 mm, layer height to 0.1 mm, infill to 100%, and print speed to 200 mm/s. We also found that the concentric infill pattern results in prints that more closely match our simulation model (Figure 5).

4.2 Image-Based Release Measurement

Measuring the dissolution process is important for both the evaluation of our approach and for calibration (Section 4.3). In the pharmaceutical industry, the dissolution process is typically measured using relatively high-cost, specialized dissolution testing instruments. Although devices based on viscosity gauging could be employed, we opt for an image-based setup. This is because our printing material absorbs visible light and allows for a simple, on-line measurement system using an off-the-shelf digital camera. Our setup (Figure 4a) is based on a Nikon D750 DSLR, a Walimex pro LED Siriux 160 Bi Color 65W LED panel, Fisherbrand 600 mL Boro 3.3 beakers (FB33114), and a Heidelberg Instruments MR Hei-Tec 145 mm diameter hotplate. We use distilled water from Dr. Starke® as our solvent. The LED panel is oriented parallel to the image plane and positioned such that its center coincides with the image center. An additional diffusing board is placed in front of the LED panel. We turn on the light and the camera around 30 minutes before the capture to let them reach a steady state. We fill the beaker with 500 g of water and cover it with a silicone-edged lid to prevent evaporation. Throughout the experiment, we maintain the water at a constant temperature (70°C) and use a stirring magnet (at 150 rpm) to homogenize the solution. We place the stirring magnet inside a hand-designed holder that reduces turbulence. We capture an image of the water to use as reference and then drop the object inside the beaker. We record images of the dissolution process every 11 seconds using an f/22 aperture, 1/125 s exposure, and native ISO 160. We stop the capture when we observe the concentration to be steady. The release curve measurement (including heating the water to 70°C) can take up to 3-4 hours depending on the shape of the object.

We map the intensity of the captured images to the corresponding concentrations of the solution. We find this mapping through a fitting procedure. We dissolve known quantities of the filament in a fixed amount (500 g) of water and take pictures once the solution is homogeneous. We add pieces of filament to the solution incrementally, waiting for each to completely dissolve before adding the next. As shown in Figure 4b, we select a square crop region of the image inside the beaker and then subdivide this area into a 10x10 grid. To safeguard against possible light variation, we normalize the pixels within the grid by dividing their intensities channel-wise by the intensities of their mirrored counterparts outside the beaker (mirroring across a vertical axis running through the image center). We then average the normalized values over each square, obtaining for each square a single normalized transmittance value per capture. Then, separately for each square, we fit an exponential function to the series of known concentration and normalized transmittance pairs. Our choice of an exponential function is motivated by the negligible scattering of the filament, meaning that Beer-Lambert’s law is applicable. The result is a separate map from known concentrations to transmittances for each square that we invert and then average to obtain the final mapping from transmittance to concentration. Images can be noisy due to capture errors or intrusions of the object into the crop region, which manifests as noise in the measured release curves; we employ a Savitzky–Golay filter to smooth the curves [Savitzky and Golay 1964].

4.3 Calibration

Our measurements and simulations of dissolution have different time and concentration units. While measurement is done in physical units (g/kg and seconds), the simulation uses non-physical units for both time and concentration. Calibrating the system consists of building a mapping between these units. For this purpose, we print a set of different shapes of equal mass, measure their release profiles, and simulate them using our forward model. We assume that there exist two decoupled linear mappings between the physical concentration and time units and their simulation counterparts. The scalar coefficient $\tilde{C}$ of the linear map for concentration is given simply by the ratio of the final solution concentration in the measured release curves; we employ a Savitzky–Golay filter to smooth the curves [Savitzky and Golay 1964].

$$\mathcal{L}(C_s, C_r, k) = \int \omega_s |k C_s^r(kt) - C_r^s(t)| + |C_s^s(kt) - C_r^r(t)| dt,$$

where $C_s$ and $C_r$ are the cumulative release profiles in simulation time and in real time, respectively, and weight parameter $\omega_s$ (set to 1 in our implementation) adjusts the relative importance of fitting the curve slopes.
We first evaluate the accuracy of our forward model (in conjunction with physical measurements) and compare it with a standard Monte Carlo approach. Then, we put our inverse design pipeline to work, showing an extensive set of experiments both in simulation and with physical measurements.

5 EVALUATION

5.1 Forward Model Evaluation

We evaluate the accuracy of our forward model by fabricating 5 objects with different shapes but roughly the same mass (∼1.75 g). For each model, we measure and simulate their release profiles. The simulated release profiles are converted into physical units using our calibration step (Section 4.3) and shown side-by-side with measured release curves in Figure 8. We follow a cross-validation approach where the measurement and simulation data of each individual shape is excluded for computing its calibration. We also include the results of the Monte Carlo simulation, which has been used in the past for forward modeling of the release of 3D printing materials [Nisser et al. 2019]. We tune two important parameters of the MC simulation, namely the pre-exponential factor and the activation energy in the Arrhenius equation, by searching over a grid of 16 combinations [Pauling 2014]. Figure 8 shows the results of the forward model evaluation. The release curves simulated by our method show good agreement with the measured curves—on par with or better than the MC predictions. Apart from its accuracy, our method is distinguished from MC by its differentiability.

5.2 Inverse Design Evaluation

In order to factor out the effects of calibration, our first set of inverse design validation experiments starts from a source shape, simulates its release profile using our forward model, and sets that profile as the target $C^*(\tau)$ for inverse design. For each target, we run three variants of our inverse method with: no fabricability constraint (NFC), the extrusion of a 2D density field (Extruded), and a fully 3D solution with incremental compliance (ICO) penalized by $J_{\text{fab}}$. Figure 6 shows the resulting set of diverse inverse-designed geometries. We observe that our inverse method in all of its variants is able to accurately fit to the target release curve. Note that the plotted release profiles are a result of a full forward simulation of the solid geometries shown in Figure 6 and not simply the release curve evaluated on the density field found by inverse optimization. Specifically, we extracted the $\rho = \frac{1}{2}$ contour of the optimal density field and then voxelized the resulting solid into a purely binary design for simulation.

The calibration step also compensates for mismatches between our digital designs and their fabricated counterparts. While our forward model assumes a homogeneously filled, isotropic solid, the parts fabricated using FDM exhibit prominent infill-dependent heterogeneities and highly anisotropic properties aligned with the filament direction. In this work, we use the concentric infill pattern for printing all objects as it yielded the highest density. Figure 5 shows a single geometry printed with various infill patterns and measured using our image-based setup. We compare all measurements against the simulated release curve of the digital model, which has been transformed into physical units using our calibration (performed on a set of objects printed with concentric infill). Thanks to the calibration, the measured release profile of the print with concentric infill matches the simulation significantly better than the other infill prints’ curves.

The calibration step also compensates for mismatches between our digital designs and their fabricated counterparts. While our forward model assumes a homogeneously filled, isotropic solid, the parts fabricated using FDM exhibit prominent infill-dependent heterogeneities and highly anisotropic properties aligned with the filament direction. In this work, we use the concentric infill pattern for printing all objects as it yielded the highest density. Figure 5 shows a single geometry printed with various infill patterns and measured using our image-based setup. We compare all measurements against the simulated release curve of the digital model, which has been transformed into physical units using our calibration (performed on a set of objects printed with concentric infill). Thanks to the calibration, the measured release profile of the print with concentric infill matches the simulation significantly better than the other infill prints’ curves.
Fig. 6. Inverse designs evaluated in simulation. Starting from a source shape (a), we simulate its release curve (solid green line in (b)) and optimize for release-equivalent shapes (c-e) with different fabricability constraints. The resulting release curves are plotted in dashed lines in (b). All axes use simulation units.
Apart from the high accuracy in matching the release profiles (apparent in Figure 6b), it is interesting to see the release equivalency between wildly different shapes. This observation highlights the large degree of multi-modality of the shape-from-release problem. The first row, sphere, is a simple validation where, when not constrained, our method recovers the sphere shape from our uniform initialization. Note also how the ICO constraint helps obtain a shape that does not require support by avoiding problematic overhang angles; the NFC solution without this term features unprintable overhangs. The extruded solutions show high accuracy given that they are fast to compute and lend themselves to a wide range of manufacturing methods. We note that the release of the extruded designs match those of sources with extruded geometries (P11 and CSG-1) perfectly. This is because the extrusion constraint regularizes the problem very well. However, the extruded design space
lacks sufficient expressiveness to emulate more complex source shapes like the Polyhedral bow and Dodecahedron geometries.

To demonstrate the performance of our designs in a real-world setup, Figure 9a shows a set of printed and measured release-equivalent shapes selected from Figure 6 based on the printability of the source geometry. Note that the computed inverse designs are in general more complex than the geometries used in our forward model evaluation (Figure 8). Except for the ICO design counterpart to the source shape CSG-1, we observe reasonable agreement between the target release curve (obtained by simulating the release of the source geometry and converting it into real units using our calibration) and the measured release of the printed geometries. However, the ICO design for CSG-1 has a significantly slower measured release behavior than predicted by its simulation (which nearly coincides with the target curve shown). We hypothesize that the main source of this inconsistency is the relatively narrow channels within this design. To verify this, we remove the channels from the mesh, print the resulting geometry, and measure its release. As Figure 10 shows, this modification results in a close match between simulation and measurement. Since we verify printing accuracy before each dissolution experiment, the problem occurs during dissolution. Due to the chemistry of our printing material, we observe that the object swells (Figure 11) inside the solution, causing its narrower channels to become blocked. While the simulation assumes that each layer in contact with water will dissolve at the same time, the blocked channels lead to locally different dissolution dynamics as water cannot flow freely even with mixing.

The previous results validated our inverse design on target release curves known to be feasible as they originate from different source shapes. However, the true power of our framework lies in discovering shapes achieving novel release behavior. In Figure 7, we test our pipeline on synthetic input release curves. While the first two curve are hand-drawn, the rest are analytical curves featuring a variety of constant-acceleration release behaviors. Of particular interest to pharmaceutical applications is the zero-acceleration, constant release rate shown by Design-4. We observe good agreement between the desired and optimized release profiles, though the extruded design space is insufficiently rich to achieve the more challenging behaviors. Imperfections in the fits for the NFC and ICO designs are introduced almost entirely by the density rounding step (contouring and re-voxelizing the continuous densities, which exhibit near-perfect fits). We note that target curves must obey a simple geometric condition to be feasible: the isoperimetric inequality. In particular, at every instant in time, the rate of release must exceed a lower bound established by the surface area of a sphere enclosing the volume remaining to be dissolved (spheres minimize the ratio of surface area to enclosed volume). Finally, in Figure 9b, we verify the performance of our inverse designs for hand-crafted input release profiles using a physical dissolution and measurement.

**Implementation details.** We initialize the optimization with uniform density, picking the spatially constant \( \hat{\rho} \) field such that its total mass equals the target final released mass, i.e. \( \int_\Omega \hat{\rho}(\hat{\rho}) \, dx = C^*(T) \). After the optimization terminates, we obtain the solid design’s boundary representation by extracting the \( \rho = \frac{1}{2} \) contour of the optimal density field. To accelerate simulation and optimization of high-resolution designs, we enforce reflectional symmetry across all three coordinate planes for the designs without fabrication constraints and across the build platform’s coordinate axes for the extruded and ICO designs. We use grids with 256 elements along each dimension and divide the design into \( L = 8 \) layers to evaluate \( f_{\text{fab}} \).

**Performance.** The computational bottlenecks of our method are the dense linear algebra operations within Knitro’s L-BFGS implementation; at \( N = 256^3 \) resolution, and leveraging reflectional symmetry, these take \( \sim 8 \) s per iteration; the fast marching, objective evaluation, and gradient computation together take \( \sim 2.5 \) s per iteration, and the elasticity solves for evaluating the ICO term take \( \sim 3.5 \) s per iteration. Inverse design using the extruded design space is significantly faster due to the smaller number of optimization variables, taking \( \sim 0.6 \) s per iteration in total. These reported
Shape from Release: Inverse Design and Fabrication of Controlled Release Structures

Fig. 9. Physical measurements of fabricated designs. In (a), the measured release curves of the 3D printed source shapes and their corresponding ICO and extruded designs from Figure 6 are plotted against the target curve used in inverse design (i.e., the simulated release of the source shape). The measured release profile of ICO CSG-1 is substantially slower than the target behavior due to the swelling of internal cavities (see Figure 10). In (b), measured releases of inverse designs from Figure 7 are compared against their hand-designed target curves and their simulated release.

Fig. 10. Comparison of the accuracy of our forward model for a shape with internal channels and the same shape with filled channels. We note that in the physical dissolution process, the inner channels are blocked by swelling material, leading to a mismatch with our prediction.

Hyperparameter tuning. The hyperparameters of our method are: the binary and ICO term weights $w_b$ and $w_f$, the radius $r$ of the smoothing filter in voxel units, and the steepness $\beta$ of the projection filter. Typically, for each input release profile, we tune these parameters to obtain a desired shape. The most important performance criterion during tuning is fabricability as the release behavior generally can be obtained without extensive tuning. In case of extrusion, we also manually select the layer height $h$ (in voxels). We seek designs that emerge as a single connected piece and do not exhibit sharp features. In our experience, the most important hyperparameters are the weights of the binary and ICO terms. After finding reasonable settings for these two, we further tune the parameters of our projection and smoothing filters. We have summarized the hyperparameters used for our designs in Tables 1 and 2. Moreover, Figure 12 demonstrates designs that are release-equivalent to the CSG-1 object but optimized with different hyperparameter settings.

6 LIMITATIONS AND FUTURE WORK

Some challenging target release profiles are hard to obtain with a single material. Instead of pushing our optimized density field toward binary values, an interesting alternative would be to dither mid-range densities. For single-material 3D printing technologies, such as FDM, this can be achieved, for example, using microstructures [Martínez et al. 2018]. On the other hand, we believe our
Fig. 12. Effect of different hyperparameter settings on optimal designs. From top to bottom: the binary objective term weight $w_b$, the ICO term weight $w_f$, the smoothing filter radius $r$ (in voxels), and the projection filter parameter $\beta$.

Table 1. Hyperparameters used to compute designs shown in Figure 6.

<table>
<thead>
<tr>
<th>ICO</th>
<th>Extruded</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_b$</td>
<td>$w_f$</td>
</tr>
<tr>
<td>Sphere</td>
<td>30</td>
</tr>
<tr>
<td>Pill</td>
<td>30</td>
</tr>
<tr>
<td>Spinner top</td>
<td>20</td>
</tr>
<tr>
<td>CSG-1</td>
<td>30</td>
</tr>
<tr>
<td>Polyhedral bow</td>
<td>20</td>
</tr>
<tr>
<td>Dodecahedron</td>
<td>20</td>
</tr>
<tr>
<td>CSG-2</td>
<td>50</td>
</tr>
</tbody>
</table>

computational workflow lends itself well to designing, at least, two-material release composites, especially in a scenario where one of the materials acts as a placebo barrier. More exciting will be to design composites in which both release profiles corresponding to both materials are controlled. Our distance-based, forward model proved to be reliable enough for predicting the release behavior for a wide range of shapes under stirred conditions. A potential

Table 2. Hyperparameters used to compute designs shown in Figure 7.

<table>
<thead>
<tr>
<th>ICO</th>
<th>Extruded</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_b$</td>
<td>$w_f$</td>
</tr>
<tr>
<td>Design-1</td>
<td>40</td>
</tr>
<tr>
<td>Design-2</td>
<td>40</td>
</tr>
<tr>
<td>Design-3</td>
<td>20</td>
</tr>
<tr>
<td>Design-4</td>
<td>10</td>
</tr>
<tr>
<td>Design-5</td>
<td>20</td>
</tr>
<tr>
<td>Design-6</td>
<td>20</td>
</tr>
</tbody>
</table>

ACM Trans. Graph., Vol. 41, No. 6, Article 274. Publication date: December 2022.
approach for improving our geometric model’s accuracy is to take into account the relationship between release dynamics and other shape descriptors. For example, it is reasonable to assume that regions belonging to the same isosurface but different curvatures can dissolve differently. To implement this modification, we can benefit from research in front-propagation simulation of other physical phenomena [Sethian 1985], such as fire animation. This will, however, require non-trivial changes to our topology optimization.

7 CONCLUSION
The shape of an object can be inferred from a surprisingly wide range of properties, such as shading [Woodham 1980] or interreflections [Nayar et al. 1991]. We introduced a computational framework for computing shapes from a new physical property: its release behavior. Both our physically-inspired, but purely geometric forward modelling, and our PDE-constrained topology optimization proved to be practical solutions for forward prediction and inverse design, respectively. We observed that the problem is highly multi-modal and therefore can be regularized by different fabricability constraints depending on the fabrication technology. With the emergence of pharmaceutical drug 3D printing, we look forward to applying our method in real-world settings and with practically relevant materials.

ACKNOWLEDGMENTS
We are grateful to Serdar Durdyev, Jigyi Wu, Sebastian Cucerca, Navid Ansari, Parham Zollaghari, and Sina Ravanbodshirazi for their help with our fabrication and measurement setup.

REFERENCES

ACM Trans. Graph., Vol. 41, No. 6, Article 274. Publication date: December 2022.


Shargel et al. 2016: Our simulation is based on the Noyes-Whitney equation, the standard pharmacokinetic model for predicting dissolution of a solid [Shargel et al. 2016]:

\[
\frac{dm}{dt} = DA \frac{(C_s - C_b)}{h},
\]

where \(m\) is the dissolved mass (kg); \(D\) is the diffusion coefficient (m^2/s); \(A\) is the exposed surface area (m^2); \(C_s\) and \(C_b\) are the concentrations of the solid and bulk solvent, respectively (kg/L); and \(h\) is the thickness of the concentration gradient between the solid and the bulk solvent (m). This equation is based on Fick’s first law of diffusion, which states that the mass flux is proportional to the negative concentration gradient. Implicit in this model is an assumption that, at each instant in time, a layer with uniform infinitesimal thickness is dissolved so that the rate of dissolved mass is proportional to the exposed surface area. This means that, denoting by \(\tau\) the thickness of the cumulative layer dissolved by time \(t\):

\[
\rho A \frac{d\tau}{dt} = \frac{dm}{dt},
\]

where \(\rho\) is the mass density of the solid (kg/m^3).

We make the following simplifying assumptions to obtain a computationally efficient and differentiable geometric simulation model:

- \(D\) and \(h\) do not depend significantly on the exposed geometry (being determined primarily by the choice of solvent and the rate of mixing); and
- the solvent volume into which the geometry is dissolved is sufficiently large that \(C_b\) can be regarded as constant \((C_b(t) - C_b(t_0) = C_b)\). This assumption is not fundamental to our approach, and we analyze the effects of allowing \(C_b\) to depend on the dissolved mass in Appendix B.

Under these assumptions, we can regard the entire expression \(D\frac{C_s - C_b}{h} := \alpha\) as a constant with units kg/(m^2 s):

\[
\frac{dm}{dt} = \alpha A, \quad \frac{d\tau}{dt} = \frac{\alpha}{\rho}.
\]

We emphasize that \(\frac{dm}{dt}\) is constant in this model, meaning that the thickness of the dissolved shell \(S(t)\) grows at a constant rate.

### B CONCENTRATION-DEPENDENT RATES

If the solid dissolved is too large or the solvent volume too small, \(C_b\) cannot be treated as constant in (11). We can assume without loss of generality that \(C_b(t_0) = 0\) (if it were nonzero, we could subtract it from \(C_s\) to obtain an equivalent simulation), in which case the concentration in the solvent must come exclusively from the dissolved solid: \(C_b(t) = \frac{m(t)}{V}\), where \(V\) is the solvent volume (m^3). The assumption of a uniformly thick infinitesimal layer dissolving at each instant remains unchanged (as we still assume the solvent concentration is spatially homogeneous), and so:

\[
\frac{dm}{dt} = \frac{DA}{h} \left( C_s - \frac{m(t)}{V} \right), \quad \frac{d\tau}{dt} = \frac{dm}{d\tau}.
\]

We first apply a convenient change of variables, expressing the released mass as a function of depth rather than time:

\[
m(t) := m(\tau(t)),
\]

where \(m = \int_0^\tau m(\tau - u) \, du\) is straightforward to compute with the same distance-based approach as before. To obtain the release curve \(m\), we simply must determine \(\tau(\cdot)\), which can be done by numerically integrating the autonomous nonlinear ODE:

\[
\frac{d\tau}{dt} = \frac{dm}{d\tau} / Ap = \frac{D}{\rho h} \left( C_s - \frac{m(\tau)}{V} \right). \quad \tau(t) = 0.
\]

The derivative of the release curve objective can then be calculated as before, with an extra chain rule term accounting for the derivative of this “time remapping.”