

# Physical Property Estimation and Optimization via Constrained Latent Space Exploration

Ramana Sundararaman, , Jiqiong Qiu<sup>2</sup>, Romain Savaiano<sup>2</sup>, Matthieu Pichaud<sup>2</sup>, Maks Ovsjanikov<sup>1</sup>

<sup>1</sup> LIX, Ecole Polytechnique, IP Paris

<sup>2</sup>Danone Global Research & Innovation Center

---

## Abstract

Shape optimization is a fundamental problem in computer graphics with wide-ranging applications, including industrial design engineering, 3D generative modeling, and personalized manufacturing. While most existing generative models focus on visual appearance or geometric plausibility, they rarely aim to address structural or physical constraints that might be imposed by the industrial design considerations. On the other hand, the vast majority of literature in 3D shape optimization is geared towards single-use cases via costly physical simulation. To that end, in this paper, we propose a simple yet comprehensive technique, first for estimating a desired physical property of an object based on its 3D structure, and, secondly for generating 3D shapes with desired physical characteristics. For this, we take a data-driven approach and represent each shape as a parameterized implicit surface with an associated latent vector. To handle data scarcity, we propose a novel generative approach for new shapes by sampling within convex polytopes constructed in the latent space. This ensures that the generated shapes remain plausible while improving their physical properties. In addition to our technical contribution, we also introduce a new dataset of bottles annotated with physical properties. Through extensive experiments, we demonstrate that our method is robust and produces geometrically plausible shapes while respecting multiple prescribed constraints.

## CCS Concepts

• **Computing methodologies** → Shape analysis; Machine learning approaches;

---

## 1. Introduction

Shape optimization has been a crucial problem in the field of Computer Graphics for several decades [All02, PASS95]. The importance of this problem is underlined by the wide variety of its application areas, ranging from industrial design and 3D content generation to the design of prosthetics [Zoh18, Bjö85] to name a few. This problem is particularly challenging due to the necessity of producing designs that optimally satisfy both aesthetic and physical considerations. Moreover, the importance of shape optimization is increasing due to the evolving and multi-faceted needs of various industries. Standard practices for optimizing shape structure often strongly rely on significant design and artistic expertise in 3D sculpting, meshing and UV Layout, with significant time and human labor costs.

Traditional approaches for shape optimization often involve deterministic or stochastic search strategies, such as gradient descent or evolutionary algorithms, which strive to find the optimum shape that minimizes a predefined cost function [ZNP08, All02]. This function usually represents some form geometric criterion, like smoothness or symme-

try, but can miss *realism* in the context of a broader shape space. On the other hand, learning-based methods have been introduced to produce shapes that are visually pleasing or resemble real-world structures based on their geometric features [BRFF18b, PFS\*19, RLR\*20]. These techniques have demonstrated remarkable results, thereby contributing significantly to digital content creation.

However, both categories of approaches exhibit certain limitations. Firstly, generative models often overlook the constraints posed by *physical* or structural characteristics. In other words, they frequently generate shapes without considering their feasibility and utility in real-world conditions, thus failing to balance the intricacies of aesthetics and functionality. Secondly, among the methods that consider physical plausibility, most existing models are primarily geared towards single-use cases, such as stability under gravity [MBBO22], making it difficult to produce shapes that satisfy multiple constraints simultaneously. Finally, existing data-driven methods strongly rely on the abundance of training data for optimal performance, posing a significant barrier in real-world applications.



Figure 1: Illustration of our approach. Given an initial shape and a target topload value (53.3 annotated), we would like to find a shape for this prescribed topload (right-most). Unconstrained optimization produces a shape with significant change in the volume while optimizing in the convex neighborhood yields a more realistic shape. Ours, which optimizes in the convex-hull of a quadrilateral produces the most plausible result.

To address these shortcomings, we propose a comprehensive technique for generating 3D shapes that ensures adherence to different physical properties while maintaining realism. Our approach represents each shape as a parameterized implicit surface with an associated latent vector [PFS\*19, DZW\*20, CZ19], thus enabling a global signature for a shape from which different properties can be inferred. Given this representation, we learn a differentiable mapping between the shape signature and its various physical properties. Remarkably, we empirically show that the latent vectors constructed from implicit shape representation lead to more accurate physical property estimation compared to different categories of state-of-the-art data-driven techniques that operate on explicit shape representation. Then, for a given target physical property, we generate new shapes by sampling within convex polytopes constructed in this latent space. By navigating the “trusted part” of the latent space, we leverage the advantage of having a reduced representation and do not require abundant training data to produce plausible shapes as shown in Figure 1. Furthermore, in an effort to help advance research in shape optimization, we introduce a new dataset of consumer grade plastic bottles annotated with physical properties such as topload (stress required to collapse), point-wise density, mass, and volume. Through extensive experiments, we demonstrate that our method not only produces geometrically plausible shapes but also robustly respects multiple prescribed constraints.

In conclusion, our contributions are threefold:

1. We present a novel, comprehensive technique for generating 3D shapes that adhere to multiple physical properties.
2. We propose to exploit parameterized implicit surfaces with associated latent vectors and sampling within convex polytopes in the latent space for shape generation.
3. We introduce a new dataset of shapes with annotated physical properties for shape analysis and optimization.

## 2. Related Work

### 2.1. 3D Deep Generative Models

A wide range of approaches have been proposed in recent years for 3D shape synthesis. The use of generative models for voxel grids [WZX\*16, SM17, GFRG16, GJvK20] represents a natural progression from the significant advancements made in image generation problems. However, this approach encounters challenges related to high computational costs, which impede the generation resolution and overall quality. To address this issue, some studies have proposed a more efficient shape representation based on octrees [TDB17, RUG17] to mitigate the demanding memory requirements. Nevertheless, even this sparse representation has limitations in terms of resolution and fails to capture the intricate details of 3D shapes. In an effort to enhance generation quality, researchers have explored alternative shape representations, including point clouds [ADMG17, HHGCO20], surface meshes [TGLX18, GFK\*18, WZL\*18, GYW\*19], multi-view depth maps [ASHW\*17], implicit functions [CZ19, PFS\*19, KFW20], among others.

However, most of these methodologies focus on low-level geometry, neglecting the inherent shape structure during the generation process. The spatial arrangements of objects and relationships between components have been recognized as critical for establishing and understanding structural information [MWZ\*13, CRXZ19]. Recently, several studies advocate for incorporating shape structure alongside geometry in the learning process. Nash and Williams [NW17] suggest generating segmented 3D objects part-by-part, while Li et al. [LXC\*17] and Mo et al. [MGY\*19] introduce generative neural networks for 3D structures, represented as binary trees and  $N$ -ary hierarchies, respectively. In contrast, the work in [WWL\*19, WZX\*20] views 3D shapes as a sequence of part geometries. The method proposed in [KGHB19] takes a step further by learning primitive abstraction to enhance the understanding and synthesis of 3D shapes. Shapes are

synthesized with part labeling in [WSH\*18], and another set of techniques generates 3D shapes by composing parts, as seen in [SKZCO19, YCC\*20, DXA\*19]. Similarly, Mo et al. [MWYG20] utilize a tree-hierarchy representation from [MZC\*19] to generate 3D shapes.

Related to our approach are also techniques that promote physical properties of generated 3D shapes. For example, in the work presented by [GYW\*19], the authors introduce a deep generative neural network designed to generate structured deformable meshes while ensuring support stability. Additionally, they put forth an optimization pipeline that leverages the inferred support relationships to enhance the results, achieving physically stable and well-connected shapes.

Another approach, [SCS\*20] enhances the quality of the generated results by iteratively filtering the generated content and injecting the filtered content back as training data.

More recently several approaches have been proposed for incorporating physical constraints in data-driven approaches [MBLO21, MBBO22, YYZ\*23, LWY23] aided, in part, by advances in differentiable simulation [HAL\*19]. For example, in [MBBO22] the authors explicitly embed physical constraints into the training objective function. The physical understanding is hence explicitly derived from the objective function rather than implicitly from the data. This leads to a better control of the physical quality but also relies on the availability of a differentiable simulator capable of modeling the desired property.

## 2.2. Shape Optimization

Our work is closely tied to techniques in shape optimization, which is a well-established field with a rich history rooted in applications spanning structural mechanics to electromagnetism [All12, PL15]. Originating from applications in digital fabrication, shape optimization problems have also been explored in computer graphics. The goal is to identify shape variations that satisfy specific design objectives, encompassing physical properties like stability [PWLSH13, WW16, ZXZ\*17], rotational dynamics [BWBSH14], structural stability and durability [UIM12], and constraints related to aerodynamics and hydrodynamics [BRFF18a].

In contrast to these approaches and aligning more closely with our methodology, the authors in [BRFF18a] employ a neural network to formulate their optimization objective function. They train a Geodesic Convolutional Neural Network [MBM\*17] to construct a differentiable fluidynamics simulator, subsequently utilized to optimize input shape parameters. Diverging from prior works, our proposal involves operating on the latent representation of the shape and entails the development of both a physical property estimator and, more importantly, *constrained* navigation in the latent space, which produces more plausible shapes without relying on prohibitive amounts of training data or the presence of a differentiable simulator.



Figure 2: Examples of shapes of various bottles in our dataset.

## 3. Motivation and Overview

Our main goal is to develop and evaluate machine learning-based approaches, first, for *estimating* physical properties of 3D shapes based on their geometric structure, and second, for *generating* plausible shapes with prescribed physical characteristics.

To achieve this goal, we first assemble an annotated dataset, which contains a wide variety of 3D designs of bottles annotated with three main properties, namely, topload (described in Sec. 4.2), volume and mass. The details of our dataset are provided in Section 4. Then, we compare various differentiable representations of surfaces via implicit functions in Section 5.1. Subsequently, equipped with an annotated dataset and implicit surface representation, we describe our approach for *inferring* this physical property in Section 5.2. Finally, given these findings, we propose a novel approach for generating plausible 3D shapes via constrained latent space optimization in Section 5.4. We present our main experimental results in Section 6, while Section 7 concludes the paper, summarizing our main contributions, the limitations of our approach and future directions.

## 4. Dataset

In this section, we provide an overview of the dataset introduced for learning and evaluating different approaches.

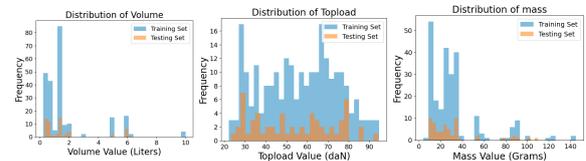


Figure 3: Distribution of various scalar properties associated with bottles that we use in shape optimization.

### 4.1. Description

As mentioned above, we evaluate our approach on a dataset consisting of consumer-grade plastic bottles represented as

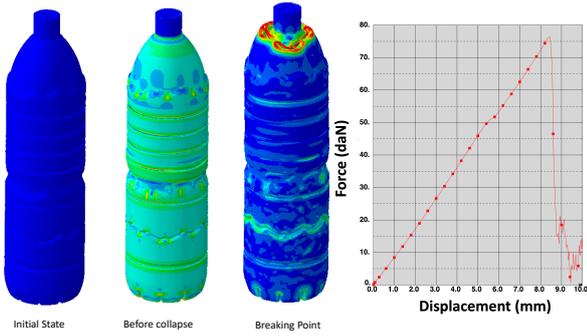


Figure 4: Depiction of how topload is estimated. Steadily increasing displacement is applied to the top of the bottle. The peak force before the bottle begins to warp (collapse) determines its topload. The plot (right) shows the measured resistance force to compression, and its *maximal value* (76daN in this case) represents the bottle’s *topload*.

triangle meshes and annotated with physical properties such as topload (maximum force resistance to vertical compression) and point-wise density. A few example shapes from our dataset are shown in Figure 2. In total, our dataset comprises of 297 triangle meshes with roughly 4000 faces per shape. We split the dataset into 80% training set and 20% as test set in accordance with standard practice. We illustrate the distribution of various physical properties in our training and test set in Figure 3. A physical property of particular interest in our dataset is the *topload* of a bottle which we describe in detail in the following section.

#### 4.2. Topload of Bottles

The key physical property we aim to optimize is the topload, which is highly relevant for objects that need to be stacked during transportation, such as bottles. The topload represents a bottle’s vertical compression strength, serving as an indicator of its pallet behavior. More intuitively, it gauges the stability of bottles, particularly plastic ones. This stability is assessed using Finite Element Analysis (FEA) techniques via Abaqus software<sup>†</sup>, where a steadily increasing displacement is applied to the top of the bottle. The peak force before the bottle begins to warp (collapse) determines its topload. Being a measure of force, topload is a scalar value per bottle, quantified in Newtons (N), but is expressed in decaNewtons (daN) in this report for simplicity. It’s worth noting that these simulations distinguish between geometry and material properties; hence, identical bottle shapes with different *masses* can have varying topload values. A numerical solver, i.e., an explicit model, is used to run the topload simulation instead of an implicit model, such as Finite Element Analysis. We used an elastoplastic PET law for the bottle’s material, with thicknesses derived from past experimental data or blowing simulations. The bottle contains both air and liquid, and the simulation is performed on

filled (non-empty) bottles. The bottle shapes are primarily designed internally and are always linked to a market or development bottle. The main design objectives are aesthetics, resistance to toplevel for pallet transport, and resistance to sideload for consumer opening.

In Figure 4, we provide a graphical representation of the behavior across different displacement values. The leftmost image shows the rest configuration of the bottle, while the second from the left shows moments before collapse occurs. The second from the right indicates reaching the breaking point, corresponding to 76 daN of force exhibited by the bottle at the moment of collapse. The graph shows the force (y-axis) exhibited by the bottle in response to the applied vertical displacement (x-axis). The *peak* of this graph is our scalar value of interest, referred to as topload. Once the bottle collapses, the force drops sharply, with the subsequent fluctuation indicating oscillatory force arising from the material. Finally, we recall that our goal in this paper is to generate a bottle such that its corresponding topload matches the specified target value.

### 5. Shape Representation, Physical Property Estimation and Optimization

In this section, our end goal is to devise a framework for conditional generation 3D shapes such that the generated shape bears the required physical property. In particular, our goal is to produce 3D shapes with a prescribed topload value, while maintaining constraints related to the volume of the shape. To that end, we first delineate how to represent each shape as a parameterized implicit surface with an associated latent vector followed by a simple data-driven technique to map these implicit latent vectors to associated physical properties using MLPs. Finally, we elaborate on our proposed shape optimization framework in the convex-hull defined by latent vectors of reference shapes.

#### 5.1. Constructing Latent Shape Representation

We first learn a high-dimensional latent representation of each shape in our training set. Rather than relying on traditional mesh-based [HHF\*19] or voxel-based [WZX\*16, HTM17] learnable embeddings, we use DeepSDF [PFS\*19], which learns a continuous latent space and leverages an auto-decoder framework to decode this latent vector into a continuous SDF. This approach facilitates various operations like shape interpolation, completion, and editing with impressive fidelity.

Given a collection of  $N$  training shapes,  $\mathcal{S}_0 \dots \mathcal{S}_N$ , we first learn their respective latent embeddings  $\lambda_0 \dots \lambda_N$  using DeepSDF [PFS\*19]. As observed in recent works, MLPs fail to produce detailed reconstructions unless trained with abundant training data [SMRO22]. To overcome this limitation and improve the expressiveness of our model, we encode the input xyz positions via Fourier Feature Mapping [TSM\*20] prior to feeding them into DeepSDF. Since

<sup>†</sup> <https://www.3ds.com/products-services/abaqus/>

sines and cosines are eigenfunctions of the Laplace operator on a regular domain, mapping the xyz coordinates to higher dimensions enables them to overcome spectral bias [TSM\*20, SMB\*20]. Thus, we can express the functionality of our network as follows:

$$f_\theta(\vec{\lambda}_i, \gamma(p)) \approx \text{SDF}(p) \quad \forall p \in \mathbb{R}^3 \quad (1)$$

Where,  $\gamma(p)$  computes the Fourier embedding for point  $p$ . It can be written as follows:

$$\gamma(\mathbf{p}) = \left[ a_1 \cos\left(2\pi \mathbf{b}_1^\top \mathbf{p}\right), a_1 \sin\left(2\pi \mathbf{b}_1^\top \mathbf{p}\right), \dots, a_m \cos\left(2\pi \mathbf{b}_m^\top \mathbf{p}\right), a_m \sin\left(2\pi \mathbf{b}_m^\top \mathbf{p}\right) \right]^\top \quad (2)$$

Where,  $a_i$  controls the amplitude and  $b_i$  controls the frequency of the Fourier series. We adapt these parameters as suggested by [TSM\*20]. Given a collection of  $N$  training shapes with ground-truth SDF for points  $p \in \mathbb{R}^3$ , the objective function for training the network is given as follows:

$$\arg \min_{\theta, \{\vec{\lambda}_i\}_{i=1}^N} \sum_{i=1}^N \sum_{j=1}^K \|f_\theta(p_j) - s_j\|_2^2 + \frac{1}{\sigma^2} \|\vec{\lambda}_i\|_2^2 \quad (3)$$

The above loss function imposes a penalty for discrepancies in the predicted signed distances with the ground truth. The evaluation points  $p_j$  are sampled close to the surface of the shape similar to [PFS\*19]. It also regularizes the norm of the latent vectors,  $\lambda_i$  weighted by  $\sigma$  to encourage compact representation. At the test time, the network parameters  $\theta$  are fixed while optimal latent code  $\vec{\lambda}_i$  is estimated by Maximum-A-Posterior (MAP) as:

$$\hat{\vec{\lambda}} = \arg \min_{\vec{\lambda}} \sum_{(p_j, s_j) \in X} \|f_\theta(\vec{\lambda}, p_j) - s_j\|_2 + \frac{1}{\sigma^2} \|\vec{\lambda}\|_2^2 \quad (4)$$

Once we have a compact latent representation of our shapes, we are now ready to describe our method for physical property prediction and shape optimization with physical property objectives.

## 5.2. Physical Property Prediction

Learning to predict the physical properties from the latent vector is an important step prior to optimization. Our approach for prediction also has to be differentiable in order to be used within a shape optimization pipeline. To this end, we use a 4-layered MLP which takes the shape latent vector, produced by DeepSDF and predicts different properties. We call this network as LP-Net, an abbreviation for Latent Physical Network. LP-Net is generic in its use and can be trained to predict one or more physical properties. While we are particularly interested in topload prediction, we also benchmark the ability of LP-Net in predicting other physical properties such as volume and weight. Since LP-Net operates on a global shape descriptor, we can simply condition the prediction of LP-Net with other physical properties of

the shape. The training objective of LP-Net can be summarised as follows:

$$\arg \min_{\phi} \|g_\phi(\lambda, \eta) - \delta\|_2^2 \quad (5)$$

Where  $\delta$  denotes the physical property that our LP-Net,  $g_\phi$  learns.  $\eta$  denotes an additional global property that our prediction can be conditioned upon. Please note that  $\eta$  is optional and including it in the learning process only shows improvement as we demonstrate in the Experiment section.

## 5.3. Physical Property Optimization

In this section, we consider the task of generating a shape bearing a topload that closely matches a query topload value. I.e, given a target value possibly starting with some initial shape, how do we obtain another shape whose topload is as close as possible to this target?

We start from a shape from the training set that bears the closest topload and perform gradient-based optimization to obtain another shape with comparable topload to match the query. While satisfying topload is a desirable constraint, we also seek to simultaneously optimize other physical properties such as volume of the generated bottle (a ‘trivial’ way to increase topload is simply to increase the shape’s volume).

To this end, we propose a constrained gradient descent such that the new generated shape not only satisfies the target topload but also produces a shape with minimal change in the volume. Each of two aforementioned tasks are further subdivided into two possible ways of performing the gradient based optimization. First is a straightforward gradient descent based approach which optimizes the given latent vector by minimizing it with the target topload and optionally controlling the volume. Recall that since the topload associated with a latent vector can be obtained via  $g_\phi$ , this step is differentiable. While this approach is simple and straightforward, such techniques often do not scale well with Deep Learning methods due to the highly non-convex nature of the optimization space. Furthermore, our goal is to also build an approach that is robust in settings where paucity of training data is a key concern. To this end, we introduce novel constrained latent-space optimization technique, where we force the optimization to be in the linear subspace defined by the convex-hull of latent vectors. By carefully choosing points that define this convex-hull, we empirically observe, as shown in the subsequent section, that our approach produces more plausible shapes. We first begin by outlining our general latent space optimization in Section 5.4 followed by the optimization in the convex-hull defined by training shapes in Section 5.5. Finally, we show that incorporating the additional constraint on the physical properties can be done in a straightforward manner to both class of aforementioned optimization in Section 5.6.

## 5.4. Generic Latent Space optimization (LSO)

Using the pre-trained LP-Net and DeepSDF, the goal is to sequentially modify the latent vector to generate new shapes

which are realistic and at the same time faithfully reflect the change in physical property of the object of interest. Let  $f_\theta$  denote trained Deep-SDF with fixed parameters  $\theta$ ,  $g_\phi$  denote the LP-Net with fixed parameters  $\phi$  and the latent vector denoted by  $\vec{\lambda}$ . Then, the task of LSO is to find the optimal latent vector which minimizes the following energy:

$$\mathcal{E} = \operatorname{argmin}_{\vec{\lambda}} \Lambda_1 \sum_{i=1}^N |f_\theta(x_i, \vec{\lambda}) - s_i| + \Lambda_2 |g_\phi(\vec{\lambda}) - \delta| \quad (6)$$

Where  $\delta$  is the *scalar* physical property of interest and  $s_i$  is the signed distance value at point  $x_i$ . The parameter  $\Lambda$  controls the impact of LP-Net in modifying its weights  $\psi$ . The first term of this energy enforces geometric similarity through point-wise SDF while the second term optimises a physical property. In practice, we solve this optimization in two stages. First, we perform 800 steps of gradient descent with  $\Lambda_1 = 1, \Lambda_2 = 0$  and then perform another 800 steps by setting  $\Lambda_1 = 0, \Lambda_2 = 1$ .

### 5.5. Optimization in Convex-Hull

The constraint we imposed in the previous case is too weak in the sense that there is no guarantee that the optimized latent vector will correspond to a plausible shape. This limits the applicability especially in scenarios where there is a paucity in data. To alleviate this shortcoming, we propose to find the optimal latent vector in the region defined by convex hull of K-points. For mathematical brevity, we restrict  $K = 3, 4$  in our experiments, but theoretically K can take any value between  $K=1$  and  $K=M$ , where M is the size of training shapes. To recap,  $K=1$  was discussed in the previous section. Our objective for optimization in convex region is then defined as:

$$\mathcal{E} = \operatorname{argmin}_{\vec{\Omega}} \sum_{i=0}^N \Lambda_1 |f_\theta(x_i, \Omega) - s_i| + \Lambda_2 |g_\phi(\Omega) - \delta_1|$$

Where,  $\Omega = w_1 \vec{\alpha}_1 + w_2 \vec{\alpha}_2 + w_3 \vec{\alpha}_3 + w_4 \vec{\alpha}_4$

$$\text{s.t } w_i \in [0, 1] \text{ and } \sum_{i=1}^4 w_i = 1 \quad (7)$$

Unlike the case of optimization in Equation 6, where the latent vectors are updated through gradient descent, in this case, we only optimize the variables highlighted in green in the above equation to restrict the latent vector to lie in the convex-hull.

### 5.6. Constrained Optimization in Convex-Hull

While optimizing in the convex region implicitly imposes regularity of the generated shape, it still comes with the same flexibility as the standard LSO and allows us to add more constraints to the latent space in-terms of the scalar property we predict. I.e, we can extend the optimization for a single scalar value to multiple scalar value(s) at the same time with negligible computation cost and change to the previously established framework. In our experiments, we

use volume constraint in addition to the topload constraint. Considering  $g_{\phi_2}$  to be LP-Net with parameters  $\phi_2$  trained to predict the scalar property  $\delta_2$ , then, the dual-constraint optimization in convex region is:

$$\begin{aligned} \mathcal{E} = & \operatorname{argmin}_{\vec{\Omega}} \sum_{i=0}^N \Lambda_1 |f_\theta(\Omega, x_i) - s_i| + \Lambda_2 |g_{\phi_1}(\Omega) - \delta_1| \\ & + \Lambda_3 |g_{\phi_2}(\Omega) - \delta_2| \\ \Omega = & \sum_{i=1}^4 w_i \vec{\lambda}_i \\ \text{s.t } & w_i \in [0, 1] \text{ and } \sum_{i=1}^4 w_i = 1 \end{aligned} \quad (8)$$

Where  $g_{\phi_1}$  and  $g_{\phi_2}$  are two networks that optimise different physical properties simultaneously.

### 5.7. Implementation Details

Our code is implemented in Pytorch [PGM\*19] and we use the Framework provided by DeepSDF [PFS\*19] to amass training data with annotated SDF value. For optimization, we use ADAM [KB14] optimizer with a learning rate of  $1e - 3$ . We experimented with standard Gradient Descent and second-order method L-BFGS [LN89]. The former showed deteriorated performance while the latter did not show any improvement over ADAM. We posit the non-convex nature of this optimization to be a possible reason. For baselines that we compare with, we use the author provided code with the prescribed hyper-parameter unless stated otherwise. The weights  $w_i$  used to enforce that the latent vector remains within the convex hull must be positive fractions. Moreover, since these weights have to form the partition of unity, given a convex-hull defined by k-points, we would have to optimize for (k-1) free variables. We follow a slightly different approach for defining an arbitrary point within the convex-hull of a triangle and a quadrilateral elaborated as follows. Given  $\alpha_1, \alpha_2$  to be our optimizable variables, a latent vector  $\lambda_i$  can be expressed in the convex-hull spanned by  $\lambda_1, \lambda_2, \lambda_3$  as follows:

$$\lambda_i := w_1 \lambda_1 + w_2 \lambda_2 + (1 - w_1 - w_2) \lambda_3$$

Where,  $w_1 = \sigma(\alpha_1)(\sigma(\alpha_2) - 1)$  and  $w_2 = \sigma(\alpha_1)$

Where  $\sigma$  is the sigmoid function. Similarly, for a quadrilateral, we can express  $\lambda_i$  as follows,

$$\lambda_i := w_1 \lambda_1 + w_2 \lambda_2 + w_3 \lambda_3 + w_4 \lambda_4$$

Where,  $w_1 = \sigma(\alpha_1)\sigma(\alpha_2)$ ,  $w_2 = \sigma(\alpha_1)(1 - \sigma(\alpha_2))$ ,

$w_3 = (1 - \sigma(\alpha_2))(1 - \sigma(\alpha_1))$  and  $w_4 = (1 - \sigma(\alpha_1))\sigma(\alpha_2)$

We observed in our experiments that the above construction of the convex-hull for a quadrilateral to yield better result than using three free variables. We choose the shapes defining the convex-hull as those bearing closest toload to the target. In case we are given a query shape geometry,

we choose neighbours that has volume closest to the query shape. This distinction is better elaborated in the experiments section.

## 6. Experiments and Results

Method	CD ( $\downarrow$ )	EMD ( $\downarrow$ )
DeepSDF [PFS*19]	6.0	5.1
IM-Net [CZ19]	6.2	5.2
Curr-DeepSDF [DZW*20]	5.5	5.0
DIF-Net [DYT21]	5.3	4.8
Ours	<b>5.2</b>	<b>4.7</b>

Table 1: Comparison of different methods for implicit surface reconstruction of shapes represented as bottles. CD and EMD are scaled by  $10^3$ . We highlight the best in green and second best score using yellow.

In this section, we present qualitative and quantitative evaluation for shape representation, physical property estimation and optimization. As mentioned before, our optimization framework relies on implicit surface representation. Therefore, we first compare different possibilities for representing implicit surface in Section 6.1. Then, we will provide quantitative comparison between different surface-based and point-based methods for various physical property prediction against our approach in Section 6.2. Finally, we provide qualitative and quantitative evaluation of our novel physical property optimization framework in Section 6.3.

### 6.1. Implicit Surface Reconstruction of Bottles

We compare different approaches for implicit surface representation of bottles with our choice. To recall, our representation is the standard DeepSDF whose inputs are mapped to Fourier-Domain [TSM\*20]. We compare with 4 other methods that represents shape as a latent vector and encodes the Signed Distance Field into the weights of MLP. Firstly we use the two first methods to represent zero-level sets of shape via a Neural Network as our baseline, namely DeepSDF [PFS\*19] and IM-Net [CZ19]. Then we use an improved version of DeepSDF [PFS\*19] with curriculum learning strategy, referred to as Curr-DeepSDF [DZW\*20]. Our final baseline is the DIF-Net [DYT21] which extends SIREN [SMB\*20] to a collection of shapes via displacement field learnt over an inferred template. Our quantitative results are summarized in Table 1. We use Chamfer Distance and Earth Mover’s Distance as our evaluation metric, commonly used in shape reconstruction problems [ADMG17]. While being simple and computationally comparable to other baselines, our approach produces better reconstruction. We also visualize qualitative examples in Figure 5. SIREN [DYT21] preserves sharp features but at the same time produces artifacts. Our approach produces better reconstruction of finer details such as seams and creases without such artifacts.

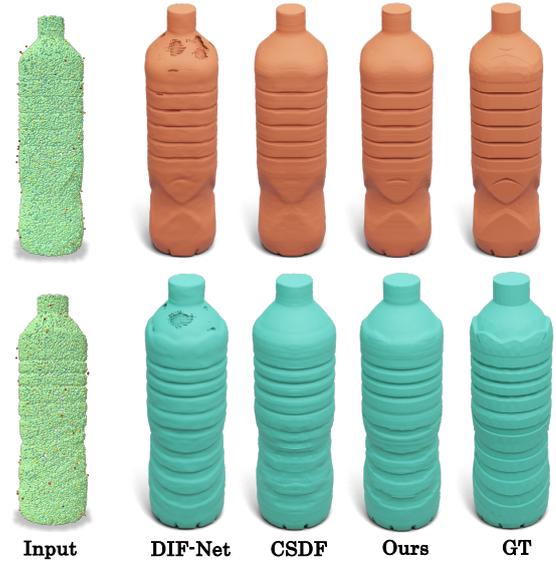


Figure 5: Comparison of implicit surface reconstruction by different methods. The first column shows the input points colored by their SDF and subsequent columns are reconstruction by different annotated methods. Ours produces an accurate reconstruction preserving detailed features such as creases

### 6.2. Physical Property Prediction of Bottles

As mentioned previously, in addition to shape representation and optimization, another goal of our work is to compare the utility of different previously proposed shape representations for estimating physical properties from the geometric shape structure. Differently from previous studies, which focus primarily on estimating semantic properties, such as shape category [WSK\*15] or part label [MZC\*19], our main goal is to evaluate some underlying *physical* property of the shape based on its geometric structure. This presents a novel application of geometric deep learning approaches, and thus, one of our contributions lies in comparing the effectiveness of different existing approaches for estimating the topload of a bottle.

#### 6.2.1. Baselines

Specifically, we compare the efficacy of our LP-NET with different categories of explicit data-driven techniques. We compare against surface-based, graph-based, and point-based methods which can leverage more principled geometric information. Namely, we use the classic PointNet [QSMG17] and PointNet++ [QYSG17] as our point-based baselines. We use DGCNN [WSL\*19] as the graph-based baseline and use DeltaConv [RAEK22] and DiffusionNet [SACO22] as surface-based baselines respectively. The latter methods have been shown to obtain superior performance compared to purely point-based methods on tasks such as shape segmentation or shape matching, as they also take into account

the surface mesh structure. However, their efficacy for *physical* property estimation is not well understood which is why we include them in our comparison. We note that the surface-based methods DiffusionNet [SACO22] and DeltaConv [RAEK22] are not fully differentiable with respect to the changes of the input geometry. This is because they rely on non-differentiable pre-processing steps, such as computing the Laplace-Beltrami eigen-decomposition [SACO22]. As such, we only include them for comparison and emphasize that they cannot be used to guide shape optimization to achieve a prescribed physical property, which is our ultimate goal.

### 6.2.2. Baseline Implementation Details

We used the author-provided implementation of different methods for our comparisons. For PointNet [QSMG17], we used two T-Nets for learning input transformations and feature transformations, each consisting of 2-layer MLPs. Between the input transformation and feature transformation, three 1D Conv layers were used. After applying a max-pooling operation to obtain a permutation-invariant feature vector, we applied a 3-layer MLP to predict topload. For PointNet++, we applied 3-layer convolutions at each scale where grouping operations were performed. Similar to PointNet, we used a 3-layer MLP over the global latent code. For DGCNN [WSL\*19], we used five learnable layers with 40 points used to construct a local neighborhood. We used Batch Normalization [IS15] in all three baselines discussed so far. For DiffusionNet [SACO22], we used xyz input with 128 eigenvalues for computing the spectral diffusion and four diffusion blocks with 128 channels. Finally, for DeltaConv [RAEK22], we scale-normalized the input point cloud and sampled 2048 points on the surface using geodesic farthest-point sampling alongside normals. The network consisted of 4-layer MLPs with 20 neighbors used for estimating the gradient.

### 6.2.3. Discussions

As outlined in Section 6.1, our approach consists of training an MLP whose input is the latent vector obtained from implicit surface representation. We train this network for a total of 300 epochs. We provide quantitative results in Table 2 where we compare the efficacy of different methods in predicting various physical properties such as topload, Volume, and Mass of bottle. While Mass and Volume are properties that can be inferred from the geometry, estimating topload also requires information on density. Therefore we compare two versions of topload prediction, namely with and without providing the mass information, with the latter denoted as topload (W/o W) in our Table. While being both simpler and computationally efficient, our approach outperforms point-based and other surface-based methods. While DeltaConv [RAEK22] produces comparable performance, using it for shape generation/optimization is not straightforward. On the other hand, our latent vectors are conducive to shape generation and therefore are better adaptable for tasks such as shape optimization.

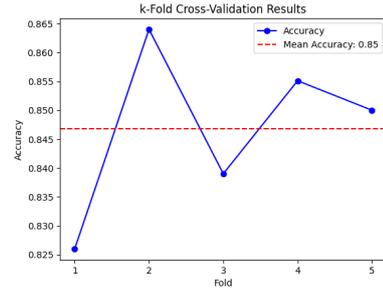


Figure 6: K-fold cross-validation plots on the accuracy of topload prediction. The observed mean accuracy is comparable to that observed on the test set.

Method	Topload	Topload (W/o W)	Vol	Mass
PointNet [QSMG17]	73.0	62.2	50.1	60.2
PointNet++ [QYSG17]	77.3	61.5	48.0	62.3
DGCNN [WSL*19]	67.7	56.4	50.8	51.8
DiffusionNet [SACO22]	73.9	59.0	54.9	57.4
DeltaConv [RAEK22]	81.2	72.1	72.0	73.1
Ours	<b>86.4</b>	<b>82.5</b>	<b>90.8</b>	<b>91.5</b>

Table 2: Comparison of our latent-vector based physical property prediction against point-based, graph-based and surface-based data-driven approaches. All numbers reported are in Percentage of accuracy.

### 6.2.4. K-fold Cross Validation

To evaluate the performance of our model in a robust and unbiased manner, we perform K-fold cross-validation. By partitioning the dataset into k-subsets (k=5), or "folds", we iteratively train and validate the model on different combinations of these folds. This approach mitigates the risk of overfitting a particular subset of data and provides a more comprehensive assessment of the model's generalizability. Additionally, K-fold cross-validation helps in identifying any variance in model performance across different data splits, offering insights into its stability and reliability. To that end, we split the training dataset into 80-20% and use the former for training and latter for evaluation. We plot the result in Figure 6. Our observed mean accuracy during the K-fold cross-validation is consistent with the mean accuracy reported on the test set, thus showing that our model does not suffer from overfitting bias.

### 6.3. Topload Optimization for Bottles

In this section, we evaluate various methods for optimizing the topload of a specified shape. Our algorithm, not predicated on a given input geometry, begins by selecting a shape (i.e., a latent vector) from the training set that approximates the desired topload. Subsequently, we refine the shape's geometry to align with the targeted topload. We examine three primary experimental scenarios. The first involves an unconstrained gradient descent, focusing solely on optimizing topload. In the subsequent two scenarios, we incorporate a

volume constraint alongside toplevel optimization. Specifically, we mandate minimal volume alteration of the shape while optimizing for toplevel. These two constrained scenarios are distinguished based on the initial proximity of the query toplevel to the target toplevel: one with a near initial condition and the other with a far initial condition. This distinction allows us to assess the efficacy of different methods in traversing the latent space and their capacity to generate varied shape configurations.

#### 6.4. Evaluation Metrics

To evaluate the effectiveness of various methods, we employ five distinct metrics: Chamfer Distance (CD), Coverage, Minimum Matching Discrepancy (MMD), and Jensen-Shannon Divergence (JSD), as originally proposed by Achlioptas et al. [ADMG17]. Coverage and Minimum Matching Distances can be calculated using either Chamfer Distance or Earth Mover’s Distance (EMD). CD is measured between the generated shape and the one possessing the target toplevel. Since these metrics necessitate geometrical evaluation, we perform this analysis using shapes from our test set. To ensure fairness, the shape corresponding to the target toplevel is withheld from all the methods under comparison.

#### 6.5. Discussions

We provide a quantitative comparison among a nearest-neighbour search, unconstrained optimization, and two forms of constrained optimization, as presented in Table 3. For the gradient descent-based methods, results are further categorized into three groups: without volume constraint, initializing with a toplevel nearest to the query, and far from the query, denoted as W/o Vol Con, NN-Init, and Far-Init, respectively. For the NN-init, we choose the shapes forming the convex-hull as those whose toplevel are closest to the target toplevel. In the Far-Init case, we choose shapes whose volumes are closest to the query shape’s volume. In the first two scenarios (W/o Vol Con and NN-Init), unconstrained gradient descent demonstrates performance comparable to that of constrained optimization. However, in the Far-Init case, unconstrained optimization significantly deteriorates, exhibiting mode collapse, a common issue in generative models [KHAK18], as highlighted by poor coverage and Jensen-Shannon Divergence (JSD) metrics. Conversely, our constrained optimization outperforms other methods and avoids this pitfall. Qualitative examples are provided in Figure 8, where the initialization is closer to the target toplevel. The first row shows that unconstrained optimization yields an unrealistic shape, divergent from the initial form, while constrained optimization results in more realistic shapes. In subsequent rows, our constrained approach consistently generates more plausible shapes than unconstrained optimization. This advantage stems from the limitation imposed by the convex-hull of known realistic shapes, proving especially beneficial in scenarios with limited training data. Additionally, Figure 9 showcases examples where



Figure 7: Illustrating the non-convergence issue due to drastic change in volume of the initial shape. Due to a sub-optimal choice of neighbor, our convex-space optimization results in an optimized shape (third column) that bears a significant volume increase.

the starting shape’s toplevel is substantially different from the target. Here, our convex optimization yields a diverse array of shapes, whereas unconstrained gradient descent often settles into local optima, producing similar outcomes for different starting shapes. Lastly, Figure 10 demonstrates the impact of enforcing a volume constraint during optimization. When a neighbor significantly differs in volume, imposing this constraint prevents the optimization from gravitating towards that shape, an effect not observed without the constraint. We demonstrate more qualitative results of shape optimization in the Appendix 8

#### 6.6. Timing

We observed that the straightforward gradient descent is the most time-efficient approach which approximately requires **12 seconds** per shape for optimization. On the other hand, the approaches based on convex hulls are slightly more expensive. Optimizing within a triangle requires **20 seconds** per shape on an average for optimization while optimization within the convex-hull spanned by a quadrilateral requires **18 seconds** per shape on an average. This is mostly because the approaches based on convex-hull also depend on the neighborhood and for the neighborhood spanned by some shapes, this optimization does not converge and has to be repeated with another pair of neighbors. We illustrate one such failure in convergence due to a drastic increase in volume in Figure 7. In our experiments, we observed convergence when one of the shapes defining the convex-hull has a comparable toplevel to one of the target toplevel. We report the average across 60 testing shapes performed on Ampere A-100 GPU.

#### 6.7. Generalization to Other Categories

The approach we have proposed for physical property estimation and optimization is general-purpose and can be adapted to different object categories. While we have demonstrated the efficacy of our approach in optimizing toplevel, a material-dependent physical property of bottles, in this section, we discuss the generalization of our methodology to a different object category and another material-dependent physical property.

Experiment	Method	CD ↓	Cov % (CD) ↑	COV % (EMD) ↑	MMD (CD) ↓	MMD (EMD) ↓	JSD ↓
W/o Vol Con	Uncon	4.2	51.0	55.6	0.87	4.2	0.017
	Trig	4.3	50.2	54.3	0.93	4.2	0.022
	Quad	<b>3.4</b>	<b>53.2</b>	<b>55.8</b>	<b>0.79</b>	<b>4.1</b>	<b>0.015</b>
NN-init	Uncon	4.1	<b>50.3</b>	48.4	<b>0.86</b>	<b>4.1</b>	0.018
	Trig	4.3	49.7	53.5	0.93	4.2	0.019
	Quad	<b>3.2</b>	48.9	<b>54.8</b>	1.10	4.5	<b>0.016</b>
Far-Init	Uncon	9.4	5.0	6.7	1.10	4.5	0.220
	Trig	4.3	31.7	31.7	1.32	4.3	0.050
	Quad	<b>3.7</b>	<b>42.0</b>	<b>58.1</b>	<b>0.92</b>	<b>4.1</b>	<b>0.039</b>
NN-search		4.8	45.7	52.7	1.00	4.4	0.110

Table 3: Summary of quantitative results for topload optimization. We report 3 scenarios, namely, without imposing volume constraint, initializing with a shape from training set of closest topload and farthest topload respectively. Arrowmarks indicate the direction of desired value. Please refer to text for more details.



Figure 8: Comparison of different techniques for optimizing the starting shape (left-most column) to match query topload. A reference shape that matches the query topload is given in the right-most column.

To that end, we consider chairs from the ShapeNetSM dataset [SCH15]. The ShapeNetSM dataset consists of approximately 12,000 objects of various categories with annotated physical properties such as weight, solid volume, sur-



Figure 9: Comparison of different techniques for optimizing the starting shape (left-most column) to match query topload. The initial shape is chosen such that its topload is furthest from the target topload. First row shows a case where we reduce the topload while the second row shows a case where topload is increased. Note that unconstrained produces a similar shape in both cases.



Figure 10: Illustration of our constrained optimization with constraint on volume of the generated bottle. Our constrained optimization produces a shape with volume close to the initialization whereas without enforcing the constraint results in a shape with disparate volume.

face volume, static friction, etc. In this section, we focus on the Chairs object class, as the total annotated data is approximately the same as for Bottles in our previous setup. We consider 'weight' as the physical property of interest to predict and optimize from the global latent representation of shape.

Consistent with our previous experimental setup, we first learn latent vectors for Chairs using DeepSDF [PFS\*19] by mapping the XYZ coordinate signal to Fourier space. Then, from these latent vectors, we learn the mapping to the anno-



Figure 11: Comparison of different techniques for generating a chair with a given query topload. Left to right: Nearest shape from training set, unconstrained optimization, optimization performed with three neighbors, optimization performed with four neighbors, and reference shape, respectively. The query topload corresponds to the reference shape.

Method	CD	Cov (CD)	Cov (EMD)	MMD (CD)	MMD (EMD)	JSD
Uncon	1.4	49.9	42.8	0.5	0.14	0.02
Trig	1.4	49.0	39.0	0.6	0.16	0.03
Quad	1.3	50.1	43.0	0.5	0.13	0.02

Table 4: Summary of quantitative results for weight optimization (increase). We provide a comparison between three baseline methods: unconstrained optimization, and two convex optimization techniques—one with two neighbors and the other with three neighbors. Please refer to the text for more details.

tated weight of the objects using our LP-Net. Finally, using the pre-trained implicit shape representation, learned latent representation, and LP-Net, we optimize for the weights of the shape. Specifically, the goal of our optimization is to produce a shape whose weight is increased by 25% compared to the initial query weight.

Since ShapeNetSM does not come with annotated property sharing correlation, we perform an unconstrained optimization (cf. Eqn 7) instead of constrained optimization as in the case of bottles. Our quantitative results are summarized in Table ???. We compare two latent convex optimization methods, namely using 3-neighbors (Trig) and 4-neighbors (Quad). The neighbors in convex space optimization have been initialized at random. In addition, we compare with unconstrained optimization (Uncon) and simply retrieving the nearest neighbor (NN-search).

## 7. Conclusions, Limitations, Future Work

In conclusion, in this work we proposed a novel a data-driven method, first to estimate and, second to optimize the physical properties of 3D objects. Our experiments underscore the robustness of our approach, which bridges the gap between geometric accuracy and physical fidelity. Interestingly, we show that previous geometric deep learning methods that operate on explicit shape representation strongly under-perform in inferring physical shape properties. In contrast, our method built upon an innovative use of neural implicit representation, provides both accurate physical property inference and enables shape optimization. By introducing a novel challenging and annotated dataset, we further contribute not only technically but also in terms of valuable resources to the research community.

Nevertheless, our work has several limitations. First, it depends on data availability for training both the latent space and our physical property estimation networks, potentially faltering with very sparse or unrepresentative datasets. The accuracy of physical property estimation, especially for intricate characteristics, also warrants improvement. Furthermore, our method’s convexity assumption in the latent space may hinder its effectiveness in scenarios where the geometry of the latent space of plausible shapes is highly complex.

In the future, we plan to investigate pre-training strategies for improved data utilization. Additionally, extending our methodology to real-time applications and developing *multimodal* optimization techniques promises to bring further improvements.

## References

[ADMG17] ACHLIOPTAS P., DIAMANTI O., MITLIAGKAS I., GUIBAS L.: Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392* (2017). 2, 7, 9

[All02] ALLAIRE G.: *Shape Optimization by the Homogenization Method*. Springer New York, 2002. 1

[All12] ALLAIRE G.: *Shape optimization by the homogenization method*, vol. 146. Springer Science & Business Media, 2012. 3

[ASHW\*17] ARSALAN SOLTANI A., HUANG H., WU J., KULKARNI T. D., TENENBAUM J. B.: Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 2

[Bjö85] BJÖRK V. O.: The development of artificial heart valves: Introduction and historical perspective. In *Guide to Prosthetic Cardiac Valves*. Springer New York, 1985, pp. 1–4. 1

[BRFF18a] BAQUÉ P., REMELLI E., FLEURET F., FUA P.: Geodesic convolutional shape optimization. In *Proceedings of the International Conference on Machine Learning (ICML)* (2018), pp. 472–481. 3

[BRFF18b] BAQUÉ P., REMELLI E., FLEURET F., FUA P. V.: Geodesic convolutional shape optimization. In *International Conference on Machine Learning* (2018). 1

[BWBSH14] BÄCHER M., WHITING E., BICKEL B., SORKINE-HORNUNG O.: Spin-It: Optimizing moment of inertia for

- spinnable objects. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 33, 4 (2014), 96:1–96:10. 3
- [CRXZ19] CHAUDHURI S., RITCHIE D., XU K., ZHANG H. R.: Learning Generative Models of 3D Structures. In *Eurographics 2019 - Tutorials* (2019), Jakob W., Puppo E., (Eds.), The Eurographics Association. doi:10.2312/egt.20191038. 2
- [CZ19] CHEN Z., ZHANG H.: Learning implicit fields for generative shape modeling, 2019. arXiv:1812.02822. 2, 7
- [DXA\*19] DUBROVINA A., XIA F., ACHLIOPTAS P., SHALAH M., GRVSCOT R., GUIBAS L.: Composite shape modeling via latent space factorization. *International Conference Computer Vision* (2019). 3
- [DYT21] DENG Y., YANG J., TONG X.: Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *IEEE Computer Vision and Pattern Recognition* (2021). 7
- [DZW\*20] DUAN Y., ZHU H., WANG H., YI L., NEVATIA R., GUIBAS L. J.: Curriculum deepsf, 2020. arXiv:2003.08593. 2, 7
- [GFK\*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B., AUBRY M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2018). 2
- [GFRG16] GIRDHAR R., FOUHEY D., RODRIGUEZ M., GUPTA A.: Learning a predictable and generative vector representation for objects. In *ECCV* (2016). 2
- [GJvK20] GUAN Y., JAHAN T., VAN KAICK O.: Generalized autoencoder for volumetric shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (June 2020). 2
- [GYW\*19] GAO L., YANG J., WU T., YUAN Y.-J., FU H., LAI Y.-K., ZHANG H.: SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019)* 38, 6 (2019), 243:1–243:15. 2, 3
- [HAL\*19] HU Y., ANDERSON L., LI T.-M., SUN Q., CARR N., RAGAN-KELLEY J., DURAND F.: DiffTaichi: Differentiable programming for physical simulation. *arXiv preprint arXiv:1910.00935* (2019). 3
- [HHF\*19] HANOCKA R., HERTZ A., FISH N., GIRYES R., FLEISHMAN S., COHEN-OR D.: Meshcnn: A network with an edge. *ACM Trans. Graph.* 38, 4 (jul 2019). doi:10.1145/3306346.3322959. 4
- [HHGCO20] HERTZ A., HANOCKA R., GIRYES R., COHEN-OR D.: Pointgmm: A neural gmm network for point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020). 2
- [HTM17] HÄNE C., TULSIANI S., MALIK J.: Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision (3DV)* (2017), pp. 412–420. 4
- [IS15] IOFFE S., SZEGEDY C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL: <https://arxiv.org/abs/1502.03167>, arXiv:1502.03167. 8
- [KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *CoRR abs/1412.6980* (2014). URL: <https://api.semanticscholar.org/CorpusID:6628106>. 6
- [KFW20] KLEINEBERG M., FEY M., WEICHERT F.: Adversarial Generation of Continuous Implicit Shape Representations. In *Eurographics Short Papers* (2020), Wilkie A., Banterle F., (Eds.). 2
- [KGHB19] KHAN S. H., GUO Y., HAYAT M., BARNES N.: Unsupervised primitive discovery for improved 3d generative modeling. In *Proc. CVPR* (June 2019). 2
- [KHAK18] KODALI N., HAYS J., ABERNETHY J., KIRA Z.: On convergence and stability of GANs, 2018. URL: [https://openreview.net/forum?id=ryepFJba-. 9](https://openreview.net/forum?id=ryepFJba-.)
- [LN89] LIU D. C., NOCEDAL J.: On the limited memory bfgs method for large scale optimization. *Mathematical Programming* 45, 1–3 (Aug. 1989), 503–528. URL: <http://dx.doi.org/10.1007/BF01589116>, doi:10.1007/bf01589116. 6
- [LWVY23] LIU X., WANG B., WANG H., YI L.: Few-shot physically-aware articulated mesh generation via hierarchical deformation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 854–864. 3
- [LXC\*17] LI J., XU K., CHAUDHURI S., YUMER E., ZHANG H., GUIBAS L.: Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)* 36, 4 (2017), to appear. 2
- [MBBO22] MEZGHANNI M., BODRITO T., BOULKENAFED M., OVSIANIKOV M.: Physical simulation layer for accurate 3d modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 13514–13523. 1, 3
- [MBO21] MEZGHANNI M., BOULKENAFED M., LIEUTIER A., OVSIANIKOV M.: Physically-aware generative network for 3d shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2021), pp. 9330–9341. 3
- [MBM\*17] MONTI F., BOSCAINI D., MASCI J., RODOLA E., SVOBODA J., BRONSTEIN M. M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 3
- [MGY\*19] MO K., GUERRERO P., YI L., SU H., WONKA P., MITRA N., GUIBAS L.: StructureNet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG, Siggraph Asia 2019)* 38, 6 (2019), Article 242. 2
- [MWYG20] MO K., WANG H., YAN X., GUIBAS L.: PT2PC: Learning to generate 3d point cloud shapes from part tree conditions. *European conference on computer vision (ECCV 2020)* (2020). 3
- [MWZ\*13] MITRA N., WAND M., ZHANG H., COHEN-OR D., BOKELOH M.: Structure-aware shape processing. In *Eurographics State-of-the-art Report (STAR)* (2013). 2
- [MZC\*19] MO K., ZHU S., CHANG A. X., YI L., TRIPATHI S., GUIBAS L. J., SU H.: Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 909–918. 3, 7
- [NW17] NASH C., WILLIAMS C. K. I.: The Shape Variational Autoencoder: A Deep Generative Model of Part-segmented 3D Objects. *Computer Graphics Forum* (2017). doi:10.1111/cgf.13240. 2
- [PASS95] PASKO A., ADZHIEV V., SOURIN A., SAVCHENKO V.: Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer* 11, 8 (Aug. 1995), 429–446. 1
- [PFS\*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: DeepSDF: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 1, 2, 4, 5, 6, 7, 10
- [PGM\*19] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KÖPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S.: *PyTorch: An Imperative Style,*

- High-Performance Deep Learning Library*. Curran Associates Inc., Red Hook, NY, USA, 2019. 6
- [PL15] PRATELLI A., LEUGERING G.: Shape Optimization in Electromagnetic Applications. In *New Trends in Shape Optimization*, vol. 166 of *International Series of Numerical Mathematics*. Springer, 2015, pp. 251–269. doi:10.1007/978-3-319-17563-8\\_11. 3
- [PWLSH13] PRÉVOST R., WHITING E., LEFEBVRE S., SORKINE-HORNUNG O.: Make It Stand: Balancing shapes for 3D fabrication. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 32, 4 (2013), 81:1–81:10. 3
- [QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 7, 8
- [QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413* (2017). 7, 8
- [RAEK22] RUBEN W., AHMAD N., ELMAR E., KLAUS H.: Deltaconv: Anisotropic operators for geometric deep learning on point clouds. *Transactions on Graphics* 41, 4 (July 2022). doi:10.1145/3528223.3530166. 7, 8
- [RLR\*20] REMELLI E., LUKOIANOV A., RICHTER S., GUILLARD B., BAGAUTDINOV T., BAQUE P., FUA P.: Meshsdf: Differentiable iso-surface extraction. In *Advances in Neural Information Processing Systems* (2020), Larochelle H., Ranzato M., Hadsell R., Balcan M. F., Lin H., (Eds.), vol. 33, Curran Associates, Inc., pp. 22468–22478. 1
- [RUG17] RIEGLER G., ULUSOY O., GEIGER A.: Octnet: Learning deep 3d representations at high resolutions. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017* (Piscataway, NJ, USA, July 2017), IEEE. 2
- [SACO22] SHARP N., ATTAIKI S., CRANE K., OVSJANIKOV M.: Diffusionnet: Discretization agnostic learning on surfaces. *ACM Trans. Graph.* 41, 3 (mar 2022). doi:10.1145/3507905. 7, 8
- [SCH15] SAVVA M., CHANG A. X., HANRAHAN P.: Semantically-Enriched 3D Models for Common-sense Knowledge. *CVPR 2015 Workshop on Functionality, Physics, Intentionality and Causality* (2015). 10
- [SCS\*20] SHU D., CUNNINGHAM J., STUMP G., MILLER S. W., YUKISH M. A., SIMPSON T. W., TUCKER C. S.: 3d design using generative adversarial networks and physics-based validation. In *Journal of Mechanical Design*, 142(7) (2020). 3
- [SKZCO19] SCHOR N., KATZIER O., ZHANG H., COHEN-OR D.: Learning to generate the "unseen" via part synthesis and composition. In *IEEE Int. Conf. on Computer Vision (ICCV)* (2019). 3
- [SM17] SMITH E. J., MEGER D.: Improved adversarial systems for 3d object generation and reconstruction. *CoRR abs/1707.09557* (2017). arXiv:1707.09557. 2
- [SMB\*20] SITZMANN V., MARTEL J., BERGMAN A., LINDELL D., WETZSTEIN G.: Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems* 33 (2020), 7462–7473. 5, 7
- [SMRO22] SUNDARARAMAN R., MARIN R., RODOLA E., OVSJANIKOV M.: Reduced representation of deformation fields for effective non-rigid shape matching. *Advances in Neural Information Processing Systems* 35 (2022). 4
- [TDB17] TATARCHENKO M., DOSOVITSKIY A., BROX T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *IEEE International Conference on Computer Vision (ICCV)* (2017). 2
- [TGLX18] TAN Q., GAO L., LAI Y., XIA S.: Variational autoencoders for deforming 3d mesh models. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (2018), IEEE Computer Society, pp. 5841–5850. doi:10.1109/CVPR.2018.00612. 2
- [TSM\*20] TANCİK M., SRINIVASAN P. P., MILDENHALL B., FRIDOVICH-KEIL S., RAGHAVAN N., SINGHAL U., RAMAMOORTHY R., BARRON J. T., NG R.: Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS* (2020). 4, 5, 7
- [UIM12] UMETANI N., IGARASHI T., MITRA N. J.: Guided exploration of physically valid shapes for furniture design. *ACM Transactions on Graphics* 31, 4 (2012), 86:1–86:11. 3
- [WSH\*18] WANG H., SCHOR N., HU R., HUANG H., COHEN-OR D., HUANG H.: Global-to-local generative model for 3d shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)* 37, 6 (2018), 214:1–214:10. 3
- [WSK\*15] WU Z., SONG S., KHOSLA A., YU F., ZHANG L., TANG X., XIAO J.: 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1912–1920. 7
- [WSL\*19] WANG Y., SUN Y., LIU Z., SARMA S. E., BRONSTEIN M. M., SOLOMON J. M.: Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)* (2019). 7, 8
- [WW16] WANG L., WHITING E.: Buoyancy optimization for computational fabrication. *Computer Graphics Forum (Proceedings of Eurographics)* 35, 2 (2016). 3
- [WWL\*19] WU Z., WANG X., LIN D., LISCHINSKI D., COHEN-OR D., HUANG H.: Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)* 38, 4 (2019), 91:1–91:14. 2
- [WZL\*18] WANG N., ZHANG Y., LI Z., FU Y., LIU W., JIANG Y.-G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV* (2018). 2
- [WZX\*16] WU J., ZHANG C., XUE T., FREEMAN W. T., TENENBAUM J. B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *arXiv preprint arXiv:1610.07584* (2016). 2, 4
- [WZX\*20] WU R., ZHUANG Y., XU K., ZHANG H., CHEN B.: PQ-NET: A generative part Seq2Seq network for 3D shapes. In *IEEE Computer Vision and Pattern Recognition (CVPR)* (2020). 2
- [YCC\*20] YIN K., CHEN Z., CHAUDHURI S., FISHER M., KIM V., ZHANG H.: COALESCE: Component assembly by learning to synthesize connections. In *Proc. of 3DV* (2020). 3
- [YYZ\*23] YANG G., YANG S., ZHANG J. Z., MANCHESTER Z., RAMANAN D.: Ppr: Physically plausible reconstruction from monocular videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 3914–3924. 3
- [ZNP08] ZINGG D. W., NEMEC M., PULLIAM T. H.: A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics* 17, 1-2 (Jan. 2008), 103–126. 1
- [Zoh18] ZOHDI T. I.: *Modeling and Simulation of Functionalized Materials for Additive Manufacturing and 3D Printing: Continuous and Discrete Media*. Springer International Publishing, 2018. doi:10.1007/978-3-319-70079-3. 1
- [ZXZ\*17] ZHAO H., XU W., ZHOU K., YANG Y., JIN X., WU H.: Stress-Constrained Thickness Optimization for Shell

Object Fabrication. *Computer Graphics Forum* (2017). doi: [10.1111/cgf.12986](https://doi.org/10.1111/cgf.12986). 3

### 8. Appendix

In this section, we show more qualitative examples of various shape optimization techniques proposed in the main article in Figure 12. An important observation we make is that unconstrained optimization produces limited shape variety, a drawback which the proposed constrained optimization does not suffer from. Especially, when the initialization is far away from the target topload, the unconstrained optimization gets “stuck” in a local minima - a latent vector corresponding to two shapes. This phenomenon can be seen highlighted in the second column of Figure 12. On the other hand, our convex optimisation produces a *blended* shape of neighbours, resulting in more than one possible shape for a given initial shape and target topload. This however could not be made possible with unconstrained optimization.



Figure 12: We show more qualitative examples of our optimization framework. Given an initial bottle (left-most), subsequent column shows the optimized result produced by different methods. Last column shows the reference bottle.