Physically-aware Generative Network for 3D Shape Modeling

Mariem Mezghanni* Malika Boulkenafed André Lieutier Maks Ovsjanikov*

LIX, Ecole Polytechnique, IP Paris*

mezghanni,maks@lix.polytechnique.fr

Abstract

Shapes are often designed to satisfy structural properties and serve a particular functionality in the physical world. Unfortunately, most existing generative models focus primarily on the geometric or visual plausibility, ignoring the physical or structural constraints. To remedy this, we present a novel method aimed to endow deep generative models with physical reasoning. In particular, we introduce a loss and a learning framework that promote two key characteristics of the generated shapes: their connectivity and physical stability. The former ensures that each generated shape consists of a single connected component, while the latter promotes the stability of that shape when subjected to gravity. Our proposed physical losses are fully differentiable and we demonstrate their use in end-to-end learning. Crucially we demonstrate that such physical objectives can be achieved without sacrificing the expressive power of the model and variability of the generated results. We demonstrate through extensive comparisons with the state-of-the-art deep generative models, the utility and efficiency of our proposed approach, while avoiding the potentially costly differentiable physical simulation at training time.

1. Introduction

3D shape generation is a central problem in both computer vision and computer graphics. The main challenge is to minimize manual intervention in the design process, while enabling the creation of new, diverse and plausible shapes. Early efforts focused on synthesizing new shapes by borrowing and assembling parts from existing collections, combining probabilistic models with geometric constraints, e.g., [23, 12, 39] among many others. More recently, deep generative methods, in particular, adversarial networks [28] and variational auto-encoders [51] have gained popularity in various applications showing promising results. However, existing works only focus on geometric, visual and structural plausibility, largely ignoring the fact that synthesized

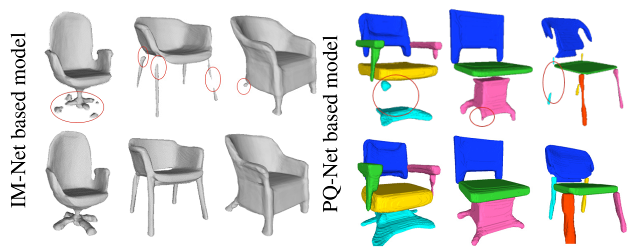

Figure 1: Visual results for 3D shape generation. We sample vectors from the latent space of IM-Net [15] and PQ-Net [71] that we decode using the corresponding baseline network (first row) and our generative network trained with the proposed physical losses (second row). Problematic regions are marked by red ovals. The resulting shapes become more connected and physically stable.

shapes are also expected to satisfy physical and functional constrains. Consequently, the generated content might appear to be a convincing example of a particular category (e.g. a chair, a car etc.) but there is no guarantee that it can be feasible and functional in the physical world. There has been a steady stream of works in the design community in studying 3D shapes from a functional perspective [36]. But, previous attempts in developing generative neural networks for unstructured [28, 51, 30] and structured [13, 44] 3D shapes have not yet jointly leveraged the power of analyzing geometric, physical and functional representations. Although it seems relatively straightforward for a human designer to make cognitive connections between geometry, physics and functionality, it is still challenging to train intelligent models to do the same.

In this paper, we introduce a physically-aware generative modeling method that makes a step to overcome these limitations (cf Figure 1). We seek a latent representation that incorporates geometric, structural and physical information. Such a latent space enables many non-trivial applications including generating novel and realistic shapes, physical shape optimization, etc. To this end, we introduce a loss that endows existing deep generative models of 3D shapes with physical reasoning.

We focus on two commonly-encountered issues in purely geometric generative models: the existence of disconnected components and the lack of stability when the object is subjected to gravity or to trivial perturbations. We demonstrate that both of these issues can be addressed through a combination of novel loss functions and a careful design of the training framework. Importantly, our approach requires no additional data or manual annotation.

Key to our approach are the implicit function representation of a 3D shape and a topological energy based on tools from persistent homology [20, 21, 80, 53, 25] coupled to promote the connectivity of the generated content. We also integrate a neural stability predictor into the generative framework to enhance the stability of generated 3D shapes when subjected to gravity. Our proposed physical loss is fully differentiable and we demonstrate its use in a variety of end-to-end learning applications. Crucially, we demonstrate that our physical objectives can be enforced without sacrificing the expressive power of the model and variability of the generated results through a careful design of the generative modeling framework.

To the best of our knowledge, our work is the first end-to-end physically-aware deep generative framework that attempts to jointly encode geometry, structure and physics in deep generative neural networks. Through extensive experiments and comparisons with the state-of-the-art deep generative networks, we demonstrate that our framework improves overall generative performance and physical plausibility metrics.

Contributions Our overall contributions are threefold. First, we demonstrate that incorporating physical reasoning as a supervisory signal into existing deep generative models can enhance the physical validity of the generated content. Second, we propose two novel learning physical losses, and explore the mutual dependency between geometry, structure and physics by encoding this information in a joint latent space. Third, we show that our framework is generalizable to different networks and 3D shape representations.

2. Related work

2.1. 3D Deep Generative Models

In recent years, 3D computer vision community has been actively investigating leveraging the power of deep generative models for 3D shape synthesis. Generative models of voxel grids [70, 61, 27, 30] constitute a natural extension of remarkable progress in image generation problems. However, this representation suffers from high computational cost that hinders generation resolution and quality. Several works propose a more efficient shape representation based on octrees [63, 57] to alleviate the prohibitive memory requirements but even this sparse representation is still limited in terms of resolution and cannot capture the fine details of 3D shapes. To improve the generation quality, researchers explored other shape representations such as point clouds [2, 33] surface meshes [62, 29, 68, 26], multi-view depth maps [5], implicit functions [15, 52, 41], etc.

The majority of these approaches, however, consider low-level geometry while discarding the shape structure in the generation process. Spatial layouts of objects and inter-part relationships are known to be useful for understanding visual information [46, 13]. Recently, several works propose to learn shape structure along with the geometry. Nash and Williams [51] propose to generate segmented 3D objects in a part-by-part manner, while Li et al. [44] and Mo et al. [47] introduce generative neural networks for 3D structures represented as binary trees and *N-ary* hierarchies respectively. In contrast, work in [72, 71] considers 3D shapes as a sequence of part geometries. The approach proposed in [40] further learns primitive abstraction to enrich 3D shape understanding and synthesis. In [66], shapes are synthesized with part labeling. Another set of methods generate 3D shapes by composing parts such as [58, 73, 19]. Similarly, Mo et al. [48] use a tree-hierarchy from [49] representation to generate 3D shapes.

The methods mentioned above do not place particular emphasis on the *physical* plausibility of the generated shapes and focus instead on the visual or structural qualities. Although several works attempt to use physical constraints such as enforcing adjacency relationships [47], they are still limited to connecting regions among shape parts.

More relevant to our work, [26] introduces a deep generative neural network that produces structured deformable meshes with support stability. They further propose an optimization pipeline that uses the inferred support relationships to refine the results to get physically stable and well connected shapes. Another work, concurrent to ours, [59] proposes to enhance the quality of the generated results by iteratively enriching the training data set with filtered generated content. In our work, we 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 and also prevents promoting certain shape structures with superior performance at the expense of the generated shapes' diversity.

2.2. Physical reasoning in deep learning

There has been increasing interest in improving generative design by exploiting physical reasoning, which forms an important signal in human-level object and scene understanding [32]. Existing works on this topic have focused on exploring physics intuition to efficiently understand 3D shapes [45, 79] and parse 3D scenes [75, 76, 18, 14]. Models from [69, 43, 7, 38, 9] were able to predict dynamics from scenes in 2D and 3D scenarios. In [78, 22], authors further consider forces and physical quantities. We refer the interested readers to a recent survey [77] on the benefits of the joint representation and joint inference of core concepts for AI with human-like common sense such as physics, causality, intents, utility, etc.

2.3. Shape Optimization

Our work is also related to efforts in shape optimization, which has a rich history motivated by applications ranging from structural mechanics to electromagnetism [3, 1]. Triggered by applications in digital fabrication, shape optimization problems have also been studied in computer graphics, aiming to find shape variations that meet certain design goals including physical properties such as stability [54, 67, 74], rotational dynamics [6], structural stability and durability [64] and aerodynamic and hydrodynamic constraints [8]. Unlike these approaches and more similar to ours, authors in [8] employ a neural network to formulate their optimization objective function. They train a Geodesic Convolutional Neural Network [50] to build a differentiable fluidynamics simulator that is then used to optimize input shape parameters. Differently from the previous works, we propose to *learn a physically-aware auto-encoder* that reconstructs an input shape while addressing the physical failures in one forward pass.

2.4. Topological regularization

Finally, our connectivity losses are based on advances in computational topology [20] and specifically on tools from persistent homology [20, 21, 80, 25, 11]. These techniques have been incorporated in applications including shape segmentation [60], 2D classification [35], 2D segmentation [37, 16], surface reconstruction [24], shape matching [53], deep learning interpretability [10], autoencoder's latent space regularization [34], etc. More relevant to ours is the work in [25] which proposes to fine-tune GAN-based generative model [70] weights using a topology layer that computes persistent homology. Instead, we propose a framework where the topological loss operates on the latent representation making it generalizable and not tied to the network architecture. We also demonstrate the effectiveness of the topological regularization for other applications that go beyond shape generation including shape auto-encoding and shape correction.

3. Method

3.1. Overview

The main idea of our work is to represent and generate shapes by jointly considering their geometry, structure and physical properties. Each shape from a particular category is believed to meet geometric consistency as well as physical constraints. In this work we focus on man-made shapes (such as chairs with attached footrest, tables, lamps, etc) that should naturally be composed of a single connected geometric component, and to be stable when subjected to gravity and to trivial perturbation forces. To exploit these two key physical cues, we introduce a connectivity loss derived from computational topology and, a stability loss based on a neural stability predictor. We then demonstrate how to endow two recent state-of-the-art generative models: the unstructured IM-Net [15] and the structured PQ-Net [71] with the proposed losses without sacrificing their expressive power. The overview of our approach is depicted in Figure 2. Given a generative network G pre-trained on a shape category, we propose to further plug a mapping function Φ at the beginning of the pre-trained generator that learns to capture physical reasoning. The idea consists of learning to map each latent vector associated with a physically implausible shape into another latent vector that represents a 'corrected' version of the same shape. Our key observation is that using this approach, the diversity of the sampled shapes is preserved since the latent space of objects is unchanged, and furthermore only latent vectors of physically implausible shapes are modified. Importantly, this approach is architecture-agnostic since it only operates on the latent space of objects.

The rest of this section is organized as follows: we first introduce our physical objective function including the connectivity and the stability losses in Sections 3.2 and 3.3. Then, we explain the network architecture in Section 3.4, and how we inject these two losses into state-of-the-art generative networks in Section 3.5. Finally, Section 3.6 describes the different applications covered by this work.

3.2. Connectivity Loss

As shown in Figures 1, 3 and 4, a common artefact in existing deep generative networks is the failure of shape connectivity. Ensuring a feasible geometry without spurious disconnected components or noise remains a challenge. To remedy this, we propose to inform the 3D shape generation through topological priors to enhance connectivity properties. To this end, we introduce a connectivity loss derived from persistent homology tools [20, 21, 80, 53, 25]. In the following, we will give a brief overview of the 0-dimensional persistent diagrams of real-valued functions and their use in our setting. We refer the interested readers to [80, 25, 53] for a more comprehensive overview.

3.2.1 Persistence diagram of real-valued function

Given a $3D$ domain $\mathcal{V} \subset \mathbb{R}^3$, we study the topological properties of a function $f : \mathcal{V} \to \mathbb{R}$. In our case, $\mathcal{V}$ is a 3D domain and f is an implicit shape representation predicted by a generative network which is defined on a finite set of points in $\mathcal{V}$ and linearly interpolated over $\mathcal{V}$. Our main fo-

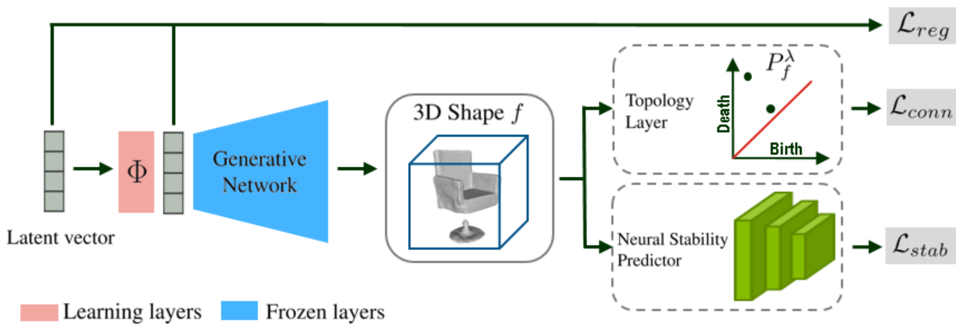

Figure 2: An Overview of our physically-aware generative network. It consists of three parts: (1) a mapping network Φ that learns to map an input latent vector associated with a physically implausible shape into another latent vector that represents a corrected version of the same shape, (2) a pre-learned generative network that decodes the latent vector into a 3D shape and (3) a physical module consisting of a topology layer and a neural stability predictor that evaluate the connectivity and stability properties respectively of the generated shapes.

cus is on the 0-dimensional homology associated with function f that reflects the number and the relative values of its local maximum (respectively minimum). For such function f, we can build a persistence diagram to track how the connected components (local maximum) of the super-levels $f^{-1}[\alpha, \infty)$ change across different values of $\alpha \in \mathbb{R}$.

To build the persistence diagram of f, we use the approach advocated by [53]. We assume that our topological space is a 3D voxel grid G. We think of each voxel as vertex and we connect each face-connected voxels by an edge. The information about the way local maximum of f evolve across decreasing α values is captured by a set of pairs (b, d) where b and d are respectively the birth and death values of each local maximum achieved by some vertex v. The birth value is simply $f(v)$ and the death value is the smallest α where $f(v) \geq f(w)$ for all w in the same connected component as v in $f^{-1}[\alpha, \infty)$. This multiset of birth-death pairs is known as the 0-dimensional persistence diagram $P_f = (b_i, d_i)_{1 \leq i \leq m}$ where m is the number of local maximum. To build P_f, we start by sorting the values of f and processing the vertices of G in descending order. A new point (b, d) is added whenever a new local maximum is detected.

3.2.2 Connectivity loss and gradient

To put this into our context, suppose f is an implicit function defined over a 3D grid where the shape surface corresponds to the λ-isosurface with f values greater than or equal to λ inside the surface. The number of connected components of this shape simply equals the number of connected components with a birth and death values in the persistence diagram P_f that are, respectively bigger and smaller than λ. We refer to this subset of the persistence diagram by $P_f^\lambda = (b_i^\lambda, d_i^\lambda)_{1 \leq i \leq m_\lambda}$ with $m_\lambda \leq m$. For notational convenience, we assume that $b_j - d_j \geq b_i - d_i$ for $j < i$. Hence, to control the connectivity of a 3D shape, we propose to optimize the following loss:

$$\mathcal{L}_c = \sum_{2 \leq i \leq m_\lambda} \left(b_i^\lambda - d_i^\lambda \right). \tag{1}$$

Note that we optimize starting from the second most persistent component $i = 2$ since our target shape is expected to have exactly one connected component. It has been shown in [53] that the derivative of $\mathcal{L}_c$ can be computed with respect to the values of f at G. The key tool is the existence of a map π that maps each pair $(b_i^\lambda, d_i^\lambda) \in P_f^\lambda$ to the pairs of vertices in G that respectively create and remove the connected component:

$$\pi : (b_i^\lambda, d_i^\lambda)_{1 \leq i \leq m_\lambda} \mapsto (v_b, v_d). \tag{2}$$

If the vertex function values are distinct (otherwise we select an arbitrary fixed vertex), then the mapping function π is unique and it yields that [25, 53]:

$$\frac{\partial \mathcal{L}_c}{\partial v} = \sum_{2 \leq i \leq m_\lambda} \frac{\partial \mathcal{L}_c}{\partial b_i} \mathbb{1}_{\pi(b_i^\lambda) = v} + \sum_{2 \leq i \leq m_\lambda} \frac{\partial \mathcal{L}_c}{\partial d_i} \mathbb{1}_{\pi(d_i^\lambda) = v}. \tag{3}$$

In the current implementation, the topological space G is a voxel grid. Consequently, values of f are evaluated at each voxel V (or equivalently vertex) of G. For implicit field representation covered by this work, $f(V)$ equals $f(x)$ for x randomly sampled inside V.

It is important to point out that authors in [25] proposed to improve the quality of a deep generative network using topological priors. Compared to their work, we avoid the triangulation of the topological space and use a cubical complex that is proved to be more efficient to study data naturally given in a cubical form [65]. Besides, we only consider a subset $P_f^\lambda \subset P_f$, since we empirically found it to improve the performance, particularly for the task of shape auto-encoding, by avoiding shape variations that don't improve the λ-isosurface parameters while preserving the connectivity optimization efficiency. Please refer to our supplementary for more details about the comparison to [25].

To sum up, we propose a novel differentiable connectivity loss that operates on 3D shape representation values $\{f(x); x \in G\}$ predicted by a generative network. $f(x)$ can be thought for instance as the distance of x to the decoded shape surface [52] for a signed distance field representation, or as the probability of x lying inside the decoded shape for voxel [70] or occupancy field [15] representations, etc.

3.3. Stability Loss

Our neural stability predictor is a neural network classifier that takes as input a 3D shape and predicts a probability $p \in [0, 1]$ that assesses whether the input is stable or not. For a given generative model G, we train a neural stability predictor Ψ_G using the corresponding generated content. To this end, we first proceed with generating a database of 3D shapes $\{S_i\}_{1 \le i \le N_G}$ using G. Then, we employ the Bullet Physics Engine [17] to simulate the behavior of each S_i when subjected to gravity and to trivial perturbation forces (please see the supplementary material for the details of the simulation settings). A shape S_i is therefore labeled stable $\{1\}$ if it remains static and $\{0\}$ otherwise. The Ψ_G architecture is tailored to the underlying shape representation; we adopt a PointNet-like [55] approach and a 3D-CNN network consisting of three convolutional layers as the base architectures to learn respectively from point cloud and voxel grid shapes. Both base architectures are followed by two fully connected layers for the classification task. We use $sigmoid$ as the activation function of the last layer in order to output the stability probability p.

3.4. Network architecture

Although our method is architecture-agnostic, in our experiments we focus on two baseline generative networks: the unstructured IM-Net [15] and the structured PQ-Net [71]. Further experiments on a 3D-VAE can be found in the supplementary material. To build our generative model, we plug a mapping network Φ consisting of four fully connected layers at the beginning of a pre-trained shape decoder, followed by our physical module (see Figure 2). For our IM-Net [15] based generator, we use the baseline decoder provided by the authors. However, for our PQ-Net [15] based one, we introduce several changes to obtain a spatial representation that we can feed to the stability loss. Specifically, we use the baseline sequential decoder followed by the part decoder which results in a sequence of voxel-based part implicit fields. Then, we additionally sample grid points that lie inside the inferred surface to which we apply the inferred part size and position transformations. The obtained point cloud is then fed to a sampling and grouping layers similarly to the work in [56] that downsample the point cloud to N=2048 points using furthest point sampling.

3.5. Training and loss function

In this section, we describe the training process of our model which is decomposed into two phases. In the first phase, we train the generative network on a dataset $\mathcal{D}$ consisting of shapes that we aim to represent and generate. In the second phase, we retrieve the decoder part of the network that we denote by G; G maps a latent vector from the learned latent space $\mathcal{V}$ into a 3D shape. Then, we augment G with our mapping network Φ inserted at the beginning of G. Φ aims to learn a mapping of each latent vector associated with a physically implausible shape into another latent vector that represents a 'corrected' version of the same shape. We denote by G' the resulting generative network from this composition $G' = G \circ \Phi$. During training, we freeze G weights and enable Φ to capture the physical reasoning enforced by the connectivity and the stability losses. At each iteration, we feed to G' a batch of sampled vectors from the pre-learned latent space of objects $\mathcal{V}$ of G. The training loss consists of three parts:

$$\mathcal{L}_{total} = \mathbb{E}_{z \in \mathcal{V}} \left[\mathcal{L}_{reg} + \alpha_c \mathcal{L}_{conn} + \alpha_s \mathcal{L}_{stab} \right], \quad (4)$$

with α_c and α_s weighting coefficients.

The regularization loss $\mathcal{L}_{reg}$: ensures the proximity to the input latent vector and thus preserves physically valid shapes and enforces $\Phi(z)$ to belong to the latent space of shapes:

$$\mathcal{L}_{reg}(z) = \|z - \Phi(z)\|_2. \quad (5)$$

The connectivity loss $\mathcal{L}_{conn}$: promotes the topological regularity of the produced shape as described in Section 3.2.2:

$$\mathcal{L}_{conn}(z) = \textstyle\sum_{(b_i^\lambda, d_i^\lambda) \in P^\lambda_{G'(z)}; i \ge 2} (b_i^\lambda - d_i^\lambda). \quad (6)$$

Note that for the part-based generative network PQ-Net [71], $G'(z)$ is a sequence of k parts expressed as $\{f_1, ... f_k\}$ where each f_j is the implicit field associated with part j. $\mathcal{L}_{conn}(z)$ equals in this case the mean of the connectivity losses computed for each part:

$$\mathcal{L}_{conn}(z) = \frac{1}{k} \sum_{f_j \in G'(z)} \textstyle\sum_{(b_i^\lambda, d_i^\lambda) \in P^\lambda_{f_j}; i \ge 2} (b_i^\lambda - d_i^\lambda). \quad (7)$$

Note that persistent diagrams are computed with a topological space G of resolution 32, except for the PQ-Net [71] based framework that uses a resolution 16 to compute each $P^\lambda_{f_j}$ for computational memory reasons.

The stability loss $\mathcal{L}_{stab}$: we use the Ψ_G neural stability predictor, trained as described in Section 3.3 to compute

$\mathcal{L}_{stab}$. Our goal is to encourage the stability probability of the $G'(z)$ to be close to one:

$$\mathcal{L}_{stab} = max(1 - \Psi_G \circ G'(z), \alpha), \quad (8)$$

with $\alpha = 0.5$ to preserve the physically stable shapes.

3.6. Applications

The latent space of our generative network provides a meaningful space for several tasks such as shape generation, auto-encoding, and correction.

Shape generation: We use latent-GAN approach [2] to sample new shapes using our physically-aware generative network. The generator and discriminator architectures consist of three fully-connected layers and trained via Wasserstein GAN loss with gradient penalty [4, 31]. At inference time, the generator maps random vectors sampled from the Gaussian distribution $\mathcal{N}(0, 1)$ into the pre-learned latent space of objects, which is then decoded into a 3D shape using our decoder.

Shape auto-encoding: We measure how accurately our network can reconstruct the encoded shapes. To build our auto-encoder, we retrieve the encoder part from each baseline network [15, 71] pre-trained each following the settings indicated by the authors. We then plug our physically aware decoder G' to reconstruct the encoded shapes. We compare the performance of our approach to baselines.

Shape Correction: This application amounts to investigating how corrupted shapes in terms of connectivity and stability can be fixed using our framework. Shape correction is performed by a simple forward pass of an input voxel shape of resolution 64 through our network consisting of a 3D-CNN encoder E_G (with similar architecture as in [15]) and our pre-trained physically-aware decoder G' based on G among [15, 71]. To learn E_G, we sampled voxel shapes of resolution 64 decoded by G, that constitute together with the corresponding latent vectors the learning set.

4. Experiments

We present quantitative and qualitative evaluations of our approach on three tasks: shape generation, reconstruction and correction. We further provide experiments on shape optimization task in our supplementary. Our experiments are conducted using PartNet dataset [49]. We use Chair category and remove shapes with more than 9 parts as described in [71] and disconnected shapes (chairs with footrest) resulting in 6253 shapes. Then we split the dataset into training/validation/test using the official split of PartNet. For fairness, all networks are trained using this setting.

4.1. Physical metrics

An important component of our work is the introduction of metrics for evaluating the physical validity of the generated content. Below, we describe the metrics we use to evaluate the quality of the results.

Connectivity: We evaluate the connectivity properties of the synthesized shapes. This evaluation is conducted on a set of sampled shapes at a resolution 256 for IM-Net [15]-based experiments and per part resolution 128 for PQ-Net [71]-based ones. For each test shape, we extract the surface using Marching Cubes which is then converted into a shape graph; where mesh nodes are thought as graph vertices and mesh edges as the graph edges. Note that we choose sampling with high resolution to have a connectivity evaluation that is correlated with the visual results, even though obviously using lower resolution of Marching Cubes would yield more connected shapes. For each sampled shape, we derive the following connectivity properties from the associated shape graph using concepts from graph theory (i) Average Connected Components (CC) counts the average number of connected components per generated shape (ii) Connection Ratio (CR): equals the number of shapes with connected graph divided by the total number of evaluated shapes (iii) Connection Ratio at 1% (CR@1) computes the connection ratio CR after removing from each shape graph connected components containing fewer than 1% of the total shape vertices. The motivation to compute CR@1 is to distinguish corrupted shapes from noisy ones.

Stability: This measure reflects the behavior of the input shape once placed on the ground in the common orientation when subjected to gravity and to trivial perturbation forces. We use two metrics (i) Average Potential Well (PW): Potential well (pw) of a stable shape equals the minimum energy needed to bring the shape into a new saddle position, beyond which the shape loses its stability. For unstable shapes under gravity, we set pw to zero. pw can be computed for each shape by studying the position of its center of mass c with respect to the support polygon $\mathcal{P}$ (defined as the convex hull of all the points of the shape touching the ground). It can be seen (see [42]) that up to a constant $pw = d - c_z$, where c_z is the height of c and d is its distance to the boundary of $\mathcal{P}$. PW is consequently the average pw values computed for the evaluation set. To compute PW, we discard shape connectivity and only focus on how shape volume is generated with respect to its support polygon. Shapes are normalized to be within a unit ball. (ii) Validity Ratio (VR) equals the number of stable and connected shapes divided by the total number of shapes to evaluate. Note that disconnected shapes are also unstable (except when disconnected parts are placed on the ground which is a rare case). The stability of the connected shapes is evaluated using the simulation process described in the supplementary material.

4.2. Physical Network training

To train our neural stability predictor Ψ_G associated with a given generative network G, we first sample multiple

G	Classifier	Accuracy
IM-NET [15]	D_G	0.544
	M_{func}	0.512
	Ψ_G	**0.964**
PQ-NET [71]	D_G	0.580
	M_{func}	0.526
	Ψ_G	**0.986**

Table 1: Quantitative stability classification results. Only Ψ_G manages to predict shape stability.

IM-Net [15]							
Net	MMD	COV	CC	CR	CR@1	PW	VR
B	7.25	51.60	2.01	58.1%	73.9%	7.32	55.5%
B+T	7.29	**51.84**	1.72	67.8%	82.8%	8.35	65.8%
B+S	**7.06**	51.35	1.68	68.7%	81.3%	**9.90**	68.5%
B+P	7.11	51.80	**1.62**	**70.9%**	**84.0%**	9.53	**70.5%**
PQ-Net [71]							
Net	MMD	COV	CC	CR	CR@1	PW	VR
B	**7.33**	57.92	2.07	53.5%	66.2%	5.37	48.7%
B+T	7.46	57.10	1.79	62.3%	73.7%	6.04	57.2%
B+S	7.38	**58.58**	2.01	56.4%	69.2%	**6.51**	53.1%
B+P	7.36	58.11	**1.74**	**65.2%**	**76.5%**	6.42	**60.5%**

Table 2: Quantitative evaluation for shape generation. B: baseline network; B+T: our network with topological loss only; B+S:our network with stability loss only; B+P: our network with both physical losses. MMD is multiplied by 10^3 and PW by 10^2.

shapes using G from which we select $N = 10K$ shapes $\mathcal{S}_G$ equally divided into stable and unstable shapes, and work with train/validation/test sets of a 80%-5%-15%. Note that $\mathcal{S}_G$ consists of voxel grid for experiments with [15] and point cloud for the ones with [71]. Furthermore, we want to draw attention to the relevance of our approach by proving that stability enhancement cannot be handled by reasoning on geometric plausibility or latent code consistency only. To this end, we perform two more experiments where we use the discriminator network described in Section 3.6 and the minimum matching distance function M_{func} as stability classifiers. D_G takes as input a 3D shape and returns a scalar; it is trained to output higher values for ground truth training shapes from [49] than for the generated ones. M_{func} takes as input a 3D shape from which we sample 2048 surface points and returns its smallest distance to our test set from [49] using Chamfer Distance. While D_G reflects the latent code consistency, M_{func} evaluates the geometric plausibility of the decoded shapes. To evaluate D_G and M_{func} in understanding 3D shape stability, we feed each test shape from $\mathcal{S}_G$ to D_G and M_{func}, then we compute the Receiver Operating Characteristic curve to find the optimal classification threshold that we use to calculate the achieved accuracy of each classifier. Table 1 displays the classification performance of each classifier on test set $\mathcal{S}_G$ for each G among IM-Net [15] and PQ-Net [71]. Results prove that D_G and M_{func} are not relevant for stability assessment and that geometric plausibility and physical stability are not necessarily correlated. Besides, we prove the feasibility of training Ψ_G to predict the stability of 3D shapes. Note that experiments with D_G and M_{func} are shown for reference and we do not intend this as a fair comparison since our method explicitly aims to detect unstable shapes.

4.3. Shape Generation

We show both qualitative and quantitative results of shape generation task in Figure 1 and Table 2 respectively. Quantitative results are performed using randomly sampled 2K shapes. We use the Minimum Matching Distance (MMD) and the Coverage (COV) to evaluate respectively the fidelity and the diversity of the generated shapes [2]. To this end, we randomly sample 2048 surface points from each generated shape and compare to our test set from [49] using Chamfer distance. As the table shows, our physical losses play a strong role. By enforcing the topological (connectivity) loss, our approach reduces noisy and disconnected components. However, this loss alone does not address stability failures. As for stability, we observe that using the stability loss only improves the connectivity for our IM-Net [15] -based generator, while it displays no improvement for the PQ-Net [71]-based one. This can be explained by the fact that the voxel-based neural stability predictor used within our IM-Net [15] -based generator helps recovering the connectivity since the stable shapes are necessarily connected (except when disconnected parts are placed on the ground which is a very rare case), while the connection property is much harder to understand when learning on point clouds, which is the case for our PQ-Net [71]-based one. Overall, we observe that jointly reasoning about connectivity and stability results in more physically valid generated shapes. Note that the common trend of performance improvement for MMD and COV metrics is also observed when combining connectivity and stability regularization. We conclude that our approach provides both meaningful and diverse outputs. Due to page limit, more qualitative results are provided in the supplementary material.

4.4. Shape Auto-encoding

In this experiment, we measure the contribution of our physical losses in reconstructing shapes from our PartNet [49] test set. We compare baselines IM-Net [15] and PQ-Net [71] with our network where we replace the decoder part G with our physically-aware decoder G'. Table 3 and Figure 3 display quantitative and qualitative evaluations respectively. We use the Intersection over Union (IoU) with sampling resolution 64, Chamfer Distance (CD) on 10K surface points with sampling resolution 256 for IM-Net [15] and per part resolution 128 for PQ-Net [71], and physical

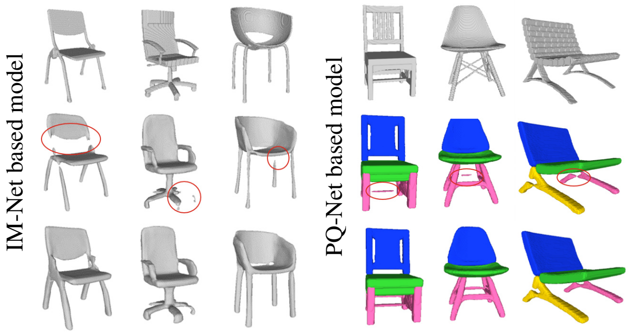

Figure 3: 3D shape reconstruction results. First row: ground truth shape; Second row: baseline network reconstruction. Problematic regions are marked by red ovals; Third row: our network reconstruction. Our approach yields better and more feasible results.

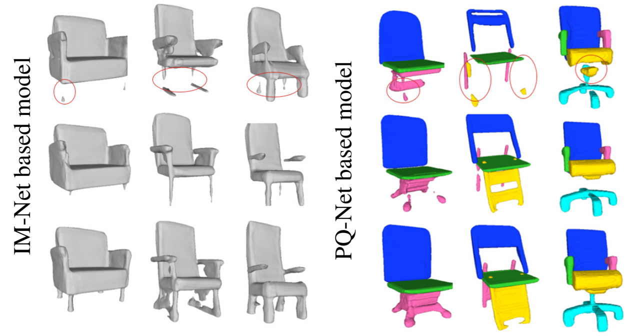

Figure 4: 3D shape correction results. First row: corrupted input shape. Problematic regions are marked by red ovals; Second row: baseline network correction. Third row: our network correction. Our approach outperforms the baselines.

Net	IoU	CD	CC	CR	CR@1	PW	VR
IM-Net [15]							
B	67.13	**2.94**	1.80	66.2%	77.7%	8.14	64.8%
B+T	70.42	3.21	1.60	71.5%	84.4%	8.55	70.2%
B+S	**70.50**	3.08	1.55	73.0%	83.4%	9.42	72.6%
B+P	70.05	3.22	**1.47**	**74.8%**	**86.4%**	**9.61**	**74.7%**
PQ-Net [71]							
B	66.08	**2.91**	1.27	84.5%	89.1%	8.24	81.9%
B+T	66.25	3.02	**1.24**	84.9%	**90.2%**	8.24	82.6%
B+S	65.00	2.98	1.25	84.8%	89.2%	**8.32**	82.3%
B+P	**66.73**	3.00	**1.24**	**85.1%**	90.0%	8.28	**82.8%**

Table 3: Quantitative evaluation for shape auto-encoding. B: baseline network; B+T: our network with topological loss only; B+S: our network with stability loss only; B+P: our network with both physical losses. CD is multiplied by 10^3, IoU and PW by 10^2.

metrics as measurements. In general, our approach outperforms the baselines in terms of physical quality. Note how the IoU is also improved, which proves that the physical correction is performed in the plausible way that reconciles differences with the underlying ground truth shape. However, we also observe an increase in CD that captures the distance between surfaces. We attribute this to the tendency of both physical losses to add volume to the input shape since it is the most frequent solution to the different failures such as connecting parts or adding legs. We aim to remedy this in future work by defining a regularization function that promotes solutions with the minimum additional volume.

4.5. Shape Correction

For each generative network G among IM-NET [15] and PQ-NET [71], we use $10K$ voxel shapes of resolution 64 to train the encoder E_G. Our test set is built by sampling and selecting $2K$ corrupted 3D shapes from each baseline G (VR equals 0.0%). Figure 4 shows several examples of input corrupted shapes that are fed to E_G then decoded by baseline G and our G' decoders. For visual convenience, we use the mesh representation of the input voxels in the provided figures. Our approach shows improvement over baseline for both G networks. Furthermore, quantitative results also support the superiority of our approach by increasing the VR from 18.9% to 34.9% using our IM-Net [15] based G' and from 37.2% to 69.8% using our PQ-Net [71] based G'. Overall, we find that physically-aware approach is considerably more accurate than the baseline in remedying connectivity and stability failures. Significantly, this improvement is obtained with a single forward pass through our decoder, and the result can be further optimized at test time.

5. Conclusion

We have proposed an approach aimed at developing a physically-aware generative neural network for 3D shapes. We demonstrated that endowing generative networks with physical reasoning can be successful to improve the generated content in terms of quality and feasibility. There are two main limitations and areas of improvement. First, our neural stability predictor relies on the training category. Although it proved its merits in predicting the stability quality of 3D shapes, we believe that shape stability is a universal cue and can be expressed in a generic and category agnostic formulation. Second, using a mapping network that operates only on latent vectors makes our approach generalizable and simple to integrate without loss of expressive power. However, extending it to retraining the weights of the generative network can further enhance the generative performance. We leave this as interesting future work.

Acknowledgements We thank the anonymous reviewers for their valuable feedback and suggestions. Parts of this work were supported by the ERC Starting Grant No. 758800 (EXPROTEA) and the ANR AI Chair AIGRETTE.

References

[1] Shape Optimization in Electromagnetic Applications. In Günter Leugering Aldo Pratelli, editor, *New Trends in Shape Optimization*, volume 166 of *International Series of Numerical Mathematics*, pages 251–269. Springer, 2015. 3

[2] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J Guibas. Learning representations and generative models for 3d point clouds. *arXiv preprint arXiv:1707.02392*, 2017. 2, 6, 7

[3] Grégoire Allaire. *Shape optimization by the homogenization method*, volume 146. Springer Science & Business Media, 2012. 3

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. 6

[5] Amir Arsalan Soltani, Haibin Huang, Jiajun Wu, Tejas D. Kulkarni, and Joshua B. Tenenbaum. 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

[6] Moritz Bächer, Emily Whiting, Bernd Bickel, and Olga Sorkine-Hornung. Spin-It: Optimizing moment of inertia for spinnable objects. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, 33(4):96:1–96:10, 2014. 3

[7] Anton Bakhtin, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick. Phyre: A new benchmark for physical reasoning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5082–5093. Curran Associates, Inc., 2019. 2

[8] P. Baqué, E. Remelli, F. Fleuret, and P. Fua. Geodesic convolutional shape optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 472–481, 2018. 3

[9] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. *international conference on robotics and automation*, 2017. 2

[10] Gunnar Carlsson and Rickard Brüel Gabrielsson. Topological approaches to deep learning, 2018. 3

[11] Mathieu Carriere, Frédéric Chazal, Marc Glisse, Yuichi Ike, and Hariprasad Kannan. A note on stochastic subgradient descent for persistence-based functionals: convergence and practical aspects. working paper or preprint, Oct. 2020. 3

[12] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *ACM SIGGRAPH 2011 papers*, pages 1–10. 2011. 1

[13] Siddhartha Chaudhuri, Daniel Ritchie, Kai Xu, and Hao (Richard) Zhang. Learning Generative Models of 3D Structures. In Wenzel Jakob and Enrico Puppo, editors, *Eurographics 2019 - Tutorials*. The Eurographics Association, 2019. 1, 2

[14] Yixin Chen, Siyuan Huang, Tao Yuan, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Holistic++ scene understanding: Single-view 3d holistic scene parsing and human pose estimation with human-object interaction and physical commonsense. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[15] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 5, 6, 7, 8

[16] James Clough, Nick Byrne, Ilkay Oksuz, Veronika Zimmer, Julia Schnabel, and Andrew King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 2020. 3

[17] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2019. 5

[18] Yilun Du, Zhijian Liu, Hector Basevi, Ales Leonardis, Bill Freeman, Josh Tenenbaum, and Jiajun Wu. Learning to exploit stability for 3d scene parsing. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1726–1736. Curran Associates, Inc., 2018. 2

[19] Anastasia Dubrovina, Fei Xia, Panos Achlioptas, Mira Shalah, Raphael Grvoscot, and Leonidas Guibas. Composite shape modeling via latent space factorization. *International Conference Computer Vision*, 2019. 2

[20] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. Applied Mathematics. American Mathematical Society, 2010. 2, 3

[21] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. volume 28, pages 454 – 463, 02 2000. 2, 3

[22] Kiana Ehsani, Shubham Tulsiani, Saurabh Gupta, Ali Farhadi, and Abhinav Gupta. Use the force, luke! learning to predict physical forces by simulating effects. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[23] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM transactions on graphics (TOG)*, 23(3):652–663, 2004. 1

[24] Rickard Brüel Gabrielsson, Vignesh Ganapathi-Subramanian, Primoz Skraba, and Leonidas J. Guibas. Topology-aware surface reconstruction for point clouds. *Comput. Graph. Forum*, 39(5):197–207, 2020. 3

[25] Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J. Guibas, and Gunnar E. Carlsson. A topology layer for machine learning. *CoRR*, abs/1905.12200, 2019. 2, 3, 4, 5

[26] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao(Richard) Zhang. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH Asia 2019)*, 38(6):243:1–243:15, 2019. 2

[27] R. Girdhar, D.F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 2

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 1

[29] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[30] Yanran Guan, Tansin Jahan, and Oliver van Kaick. Generalized autoencoder for volumetric shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 1, 2

[31] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5767–5777. Curran Associates, Inc., 2017. 6

[32] Jessica Hamrick, Peter Battaglia, and Joshua B. Tenenbaum. Internal physics models guide probabilistic judgments about object dynamics. In *In Conf. Cog. Sc.*, 2011. 2

[33] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. 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

[34] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Mandar Dixit. Connectivity-optimized representation learning via persistent homology. volume 97 of *Proceedings of Machine Learning Research*, pages 2751–2760, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 3

[35] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 1634–1644. Curran Associates, Inc., 2017. 3

[36] Ruizhen Hu, Oliver van Kaick, Bojian Wu, Hui Huang, Ariel Shamir, and Hao Zhang. Learning how objects function via co-analysis of interactions. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 35(4):47:1–47:13, 2016. 1

[37] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 5657–5668. Curran Associates, Inc., 2019. 3

[38] Carlo Innamorati, Bryan Russell, Danny Kaufman, and Niloy J. Mitra. Neural re-simulation for generating bounces in single images. *ICCV*, 2019. 2

[39] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 1

[40] Salman H. Khan, Yulan Guo, Munawar Hayat, and Nick Barnes. Unsupervised primitive discovery for improved 3d generative modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[41] Marian Kleineberg, Matthias Fey, and Frank Weichert. Adversarial Generation of Continuous Implicit Shape Representations. In Alexander Wilkie and Francesco Banterle, editors, *Eurographics 2020 - Short Papers*. The Eurographics Association, 2020. 2

[42] David Kriegman. Let them fall where they may: Capture regions of curved objects and polyhedra. *International Journal of Robotics Research*, 16(4):448–472, August 1997. 6

[43] Adam Lerer, Sam Gross, and Rob Fergus. Learning physical intuition of block towers by example. volume 48 of *Proceedings of Machine Learning Research*, pages 430–438, New York, New York, USA, 20–22 Jun 2016. PMLR. 2

[44] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4):to appear, 2017. 1, 2

[45] Zhijian Liu, William T. Freeman, Joshua B. Tenenbaum, and Jiajun Wu. Physical primitive decomposition. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[46] Niloy Mitra, Michael Wand, Hao Zhang, Daniel Cohen-Or, and Martin Bokeloh. Structure-aware shape processing. In *Eurographics State-of-the-art Report (STAR)*, 2013. 2

[47] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics (TOG), Siggraph Asia 2019*, 38(6):Article 242, 2019. 2

[48] Kaichun Mo, He Wang, Xinchen Yan, and Leonidas Guibas. PT2PC: Learning to generate 3d point cloud shapes from part tree conditions. *European conference on computer vision (ECCV 2020)*, 2020. 2

[49] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 6, 7

[50] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M. Bronstein. 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

[51] Charlie Nash and Chris K. I. Williams. The Shape Variational Autoencoder: A Deep Generative Model of Part-segmented 3D Objects. *Computer Graphics Forum*, 2017. 1, 2

[52] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 5

[53] Adrien Poulenard, Primoz Skraba, and Maks Ovsjanikov. Topological Function Optimization for Continuous Shape Matching. *Computer Graphics Forum*, 37(5):13–25, 2018. 2, 3, 4

[54] Romain Prévost, Emily Whiting, Sylvain Lefebvre, and Olga Sorkine-Hornung. Make It Stand: Balancing shapes for 3D fabrication. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)*, 32(4):81:1–81:10, 2013. 3

[55] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 5

[56] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5099–5108. Curran Associates, Inc., 2017. 5

[57] Gernot Riegler, Osman Ulusoy, and Andreas Geiger. 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

[58] Nadav Schor, Oren Katzier, Hao Zhang, and Daniel Cohen-Or. Learning to generate the "unseen" via part synthesis and composition. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019. 2

[59] Dule Shu, James Cunningham, Gary Stump, Simon W. Miller, Michael A. Yukish, Timothy W. Simpson, and Conrad S. Tucker. 3d design using generative adversarial networks and physics-based validation. In *Journal of Mechanical Design, 142(7)*, 2020. 2

[60] Primoz Skraba, Maks Ovsjanikov, Frédéric Chazal, and Leonidas Guibas. Persistence-based segmentation of deformable shapes. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, page 2146. 3

[61] Edward J. Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. *CoRR*, abs/1707.09557, 2017. 2

[62] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. 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*, pages 5841–5850. IEEE Computer Society, 2018. 2

[63] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[64] Nobuyuki Umetani, Takeo Igarashi, and Niloy J. Mitra. Guided exploration of physically valid shapes for furniture design. *ACM Transactions on Graphics*, 31(4):86:1–86:11, 2012. 3

[65] Hubert Wagner, Chao Chen, and Erald Vucini Abstract. Efficient computation of persistent homology for cubical data. Technical Report PRIP-TR-122, PRIP, TU Wien, 2010. 4

[66] Hao Wang, Nadav Schor, Ruizhen Hu, Haibin Huang, Daniel Cohen-Or, and Hui Huang. Global-to-local generative model for 3d shapes. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)*, 37(6):214:1—214:10, 2018. 2

[67] Lingfeng Wang and Emily Whiting. Buoyancy optimization for computational fabrication. *Computer Graphics Forum (Proceedings of Eurographics)*, 35(2), 2016. 3

[68] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *ECCV*, 2018. 2

[69] Jiajun Wu, Ilker Yildirim, Joseph J Lim, Bill Freeman, and Josh Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 127–135. Curran Associates, Inc., 2015. 2

[70] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems 29*, pages 82–90. Curran Associates, Inc., 2016. 2, 3, 5

[71] Ruidi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. PQ-NET: A generative part Seq2Seq network for 3D shapes. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 5, 6, 7, 8

[72] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)*, 38(4):91:1–91:14, 2019. 2

[73] Kangxue Yin, Zhiqin Chen, Siddhartha Chaudhuri, Matthew Fisher, Vladimir Kim, and Hao Zhang. COALESCE: Component assembly by learning to synthesize connections. In *Proc. of 3DV*, 2020. 2

[74] Haiming Zhao, Weiwei Xu, Kun Zhou, Yin Yang, Xiaogang Jin, and Hongzhi Wu. Stress-Constrained Thickness Optimization for Shell Object Fabrication. *Computer Graphics Forum*, 2017. 3

[75] Bo Zheng, Yibiao Zhao, Joey Yu, Katsushi Ikeuchi, and Song-Chun Zhu. Scene understanding by reasoning stability and safety. *IJCV*, 112(2):221–238, 2015. 2

[76] Bo Zheng, Yibiao Zhao, Joey C Yu, Katsushi Ikeuchi, and Song-Chun Zhu. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3127–3134. IEEE, 2013. 2

[77] Yixin Zhu, Tao Gao, Lifeng Fan, Siyuan Huang, Mark Edmonds, Hangxin Liu, Feng Gao, Chi Zhang, Siyuan Qi, Nian Ying Wu, B. Joshua Tenenbaum, and Song-Chun Zhu. Dark, beyond deep: A paradigm shift to cognitive ai with humanlike common sense. *Engineering*, 2020. 3

[78] Yixin Zhu, Chenfanfu Jiang, Yibiao Zhao, Demetri Terzopoulos, and Song-Chun Zhu. Inferring forces and learning

human utilities from videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[79] Yixin Zhu, Yibiao Zhao, and Song-Chun Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[80] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33:249–274, 02 2005. 2, 3