Physically-aware Generative Network for 3D Shape Modeling: Supplementary Material

Mariem Mezghanni* Malika Boulkenafed André Lieutier Maks Ovsjanikov*

LIX, Ecole Polytechnique, IP Paris*
mezghanni,maks@lix.polytechnique.fr

Overview

This supplementary material contains five parts:

- Section 1: provides more visual results of shape generation.
- Section 2: provides experiments on 3D-VAE model.
- Section 3: describes the settings for our simulation based stability annotation.
- Section 4: provides experiments on shape optimization task paired with gradient visualization, and comparison to several baselines.
- Section 5: provides comparison to [5].

1. Qualitative results for shape generation

In this section, we collect additional visual results about the proposed method that due to lack of space were not included in the main manuscript. Figure 1 compares the shape generation quality of baselines [3, 12] with our approach while emphasizing the importance of combining connectivity and stability losses.

2. Experiments with 3D Variational Auto-Encoder (3D-VAE)

We present further experiments using 3D-VAE to demonstrate the generalization power of our approach to different architectures and shape representations.

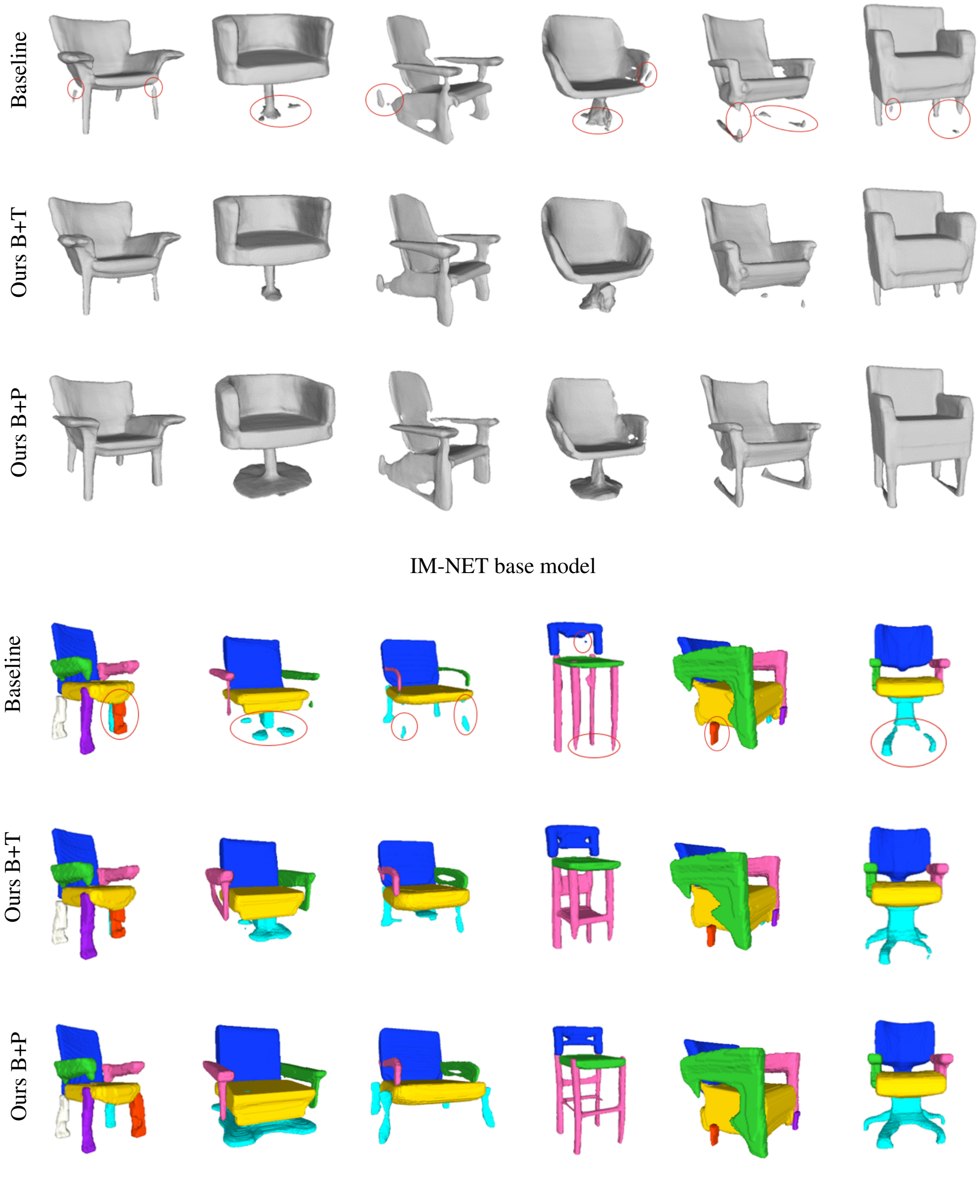
Network architecture We employ a symmetric architecture where the encoder and the decoder consist of the same number of layers. Specifically, the encoder downsamples the input shape of resolution 32 using three convolutional layers with ReLU activation, followed by a batch normalization layer. All convolutional layers have a kernel size of $4 \times 4 \times 4$, stride of 2 and filter sizes of respectively 32, 16, and 8. The output vector is then fed to two fully connected layers with Relu activation function that generates the $128D$ latent vector z. The decoder follows a similar architecture for upsampling except that we use a sigmoid activation for the last layer.

Training Similarly to the description provided in the main manuscript in Section 3.5, the training process is decomposed into two phases. During the first training phase, the network is trained using a binary cross-entropy (BCE) loss and a variational loss [8]. For the second phase, we finetune the decoder using our physical losses along with a regularization loss. Differently from the previous experiments, we update the decoder weights this time since we empirically found it to improve the results. Furthermore, just for the shape auto-encoding experiment, we train a non-variational version of our network (an auto-encoder AE) to get a higher reconstruction performance.

Results

shape generation: We demonstrate how our physical losses contribute in generating more realistic shapes. In Figure 2, we make a qualitative comparison between our technique and the baseline by randomly sampling latent vectors that we decode using the baseline decoder and our physically-aware decoder. The quantitative evaluations are given in Table 1 including physical metrics. Our approach leads to a more realistic shapes in terms of geometric quality and physical validity when combining both connectivity and stability losses. We notice how the generated shapes with our network are better connected but also realistic and plausible. Specifically, our network manages to remove noise and to add missing regions which emphasizes the complementarity between generative loss (here reconstruction and variational losses) and the physical losses.

We also demonstrate the generalization power across different categories by showing in Figure 4 results when learn-



IM-NET base model

PQ-NET base model

Figure 1: Visual results for 3D shape generation. We sample vectors from the latent space from IM-NET [3] and PQ-NET [12] that we decode using: Top row (Baseline): the baseline network. Problematic regions in baseline shapes are marked by red ovals. Middle row (Ours B+T): Our generative network with topological loss only. Bottom row (Ours B+P): our generative network with both physical losses. The resulting shapes become more connected and physically stable.

ing on Table and Airplane categories. The common trend of generation quality improvement when using physical losses holds true for different categories.

shape auto-encoding: We measure the reconstruction performance of our network and highlight the contribution of the physical losses in enriching 3D shape understanding. Table 2 and Figure 3 display quantitative and qualitative evaluations respectively. We observe that, as expected from results reported by previous experiments in the main manuscript, our approach reconstructs shapes efficiently while better preserving their physical quality.

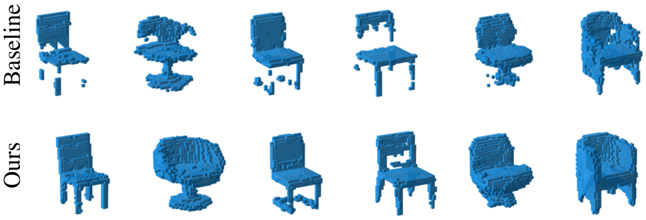


Figure 2: Visual results for 3D shape generation. We sample vectors from the latent space of our baseline 3D-VAE that we decode using (top row) baseline decoder and (bottom row) our generative network with the proposed physical losses. The resulting shapes become more realistic.

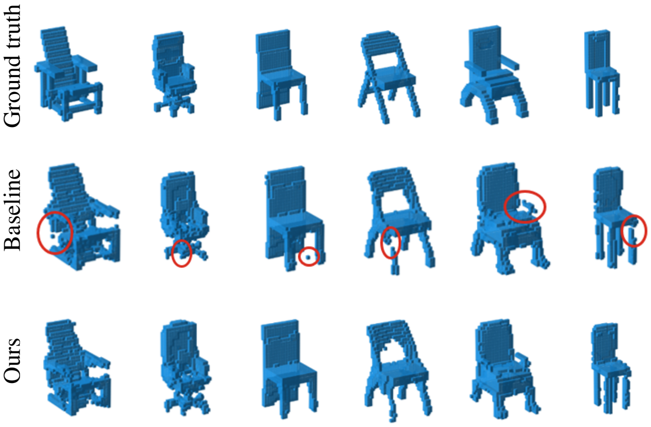


Figure 3: Visual results for 3D shape auto-encoding. Ground truth test shapes (top row) are reconstructed using the baseline AE (middle row) where problematic regions are marked with red ovals, and our AE (bottom row) where the decoder is further trained using physical losses. Physical reasoning enhances the reconstruction performance.

3D-VAE							
Net	MMD	COV	CC	CR	CR@1	PW	VR
B	8.21	49.5	4.60	20.67%	36.3%	2.84	11.8%
B+T	8.03	52.4	2.76	42.60%	64.2%	5.38	30.9%
B+S	**7.84**	**55.6**	3.25	35.40%	54.9%	**6.94**	29.0%
B+P	8.00	53.9	**2.74**	**42.63%**	**64.4%**	5.82	**31.6%**

Table 1: 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. Combining both physical losses improves the generated shapes quality.

3. Simulation Setting

In addition to the metric PW introduced in the main manuscript in Section 4.1, we also use a simulation-based method to evaluate the physical stability of 3D shapes. Note that both metrics are highly correlated. Given a set of shapes to evaluate, we can find a PW threshold above which shapes are stable according to the physical simulation and vice-versa. In our experiments where shapes are normalized within unit sphere and have equal mass and uniform volumetric mass density, this threshold approximately equals 0.01. Nevertheless, we use the physical simulation in addition to PW since it is generalizable to other physical settings and constraints.

We propose a simulation based-approach to assess a given 3D shape stability. To this end, we use PyBullet [4] as a physical engine to simulate the shape behavior when subjected to gravity and to trivial perturbation forces. Specifically, the shape and the plane are imported with URDFs including the center of mass, the mass and the inertia ma-

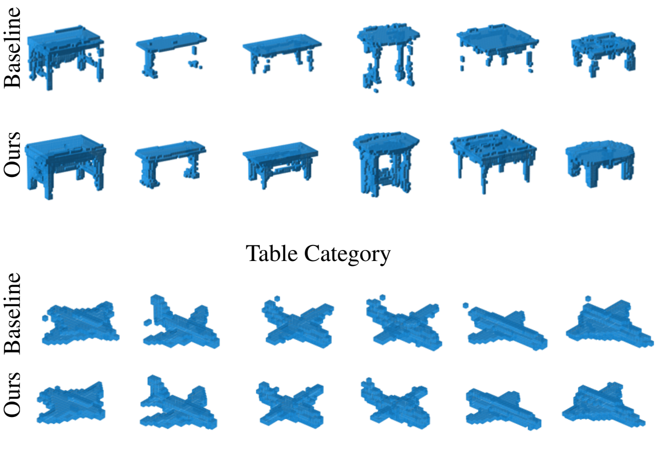


Table Category

Airplane Category

Figure 4: Visual results for 3D shape generation. We evaluate our approach for other shape categories Table (8218 shapes) from PartNet [7] and Airplane (4045 shapes) from Shapenet [2]. Note that we follow the learning process detailed in the main manuscript except for the airplane category where we only use topological loss. For each shape category, the top row shows the baseline generated shapes and the bottom row the generated shapes using our network. The generation quality improvement when using physical losses holds true for different categories.

3D-VAE						
Net	IoU	CC	CR	CR@1	PW	VR
B	73.16	1.85	65.70%	76.15%	7.75	61.70%
B+T	73.00	**1.64**	**72.12%**	**84.70%**	7.33	65.12%
B+S	**73.21**	1.85	65.95%	76.06%	**7.92**	61.99%
B+P	73.09	1.66	71.21%	81.57%	7.59	**66.18%**

Table 2: 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. IoU and PW are multiplied by 10^2. Empowering the network with physical understanding helps reconstructing realistic shapes.

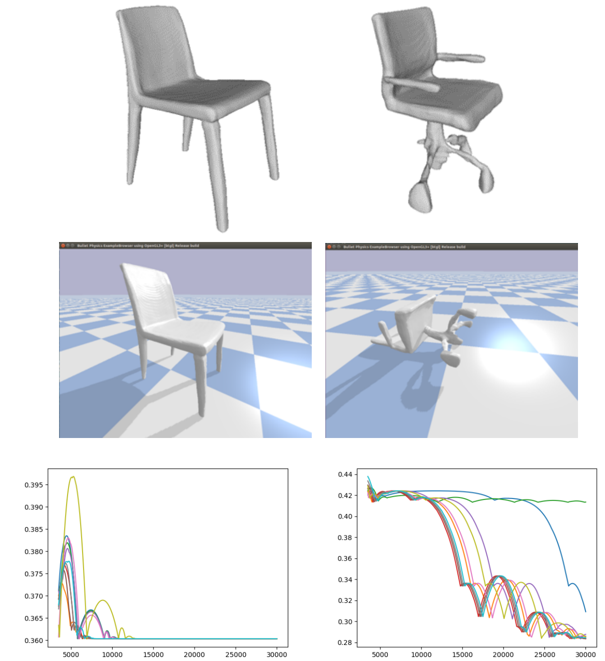


Figure 5: Simulation results. We simulate the behavior of two shapes (top row) using PyBullet [4]. The final simulation states (middle row) reflect whether the shapes are stable or not. We show the different simulation curves that record the center of mass vertical position (bottom row). Each curve corresponds to one simulation. Random perturbation forces are exerted on the shapes resulting in changing the center of mass position. Stable shape (left) manages to recover its initial state (label= 1), while unstable shape (right) loses its stability and fall (label= 0).

trix. We use the default Coulomb friction model and consider a coefficient of bouncyness for the shape plane contact equal to 0 (inelastic). Note that the generated 3D shapes are pre-aligned to the common up direction and vertical facing direction. We simulate the dropping of each shape from a height equal to 0.1 times shape height for 3000 time steps of $0.008s$ while applying small perturbations. If the shape doesn't recover its initial orientation, then it is is unstable (label=0), otherwise the shape is labeled stable (label=1). (see Figure 5). We combine this binary metric with PW to get a complete understanding and fair comparison of physical stability quality.

We recall that the physical simulation is used to annotate the learning set of the neural stability predictor and to compute VR (see Sections 3.3 and 4.1 respectively in the main paper).

4. Single Shape Optimization

Method. Note that our approach aims at generating plausible shapes in a single forward pass. Here, we consider the particular task of single shape optimization to both explain the functioning of our physical losses and to compare the latter to concurrent approaches. Using notations similar to that of the main paper, we randomly sample latent vectors $z_0 \in \mathcal{V}$ and optimize z_0 to obtain physically plausible shape. Formally, we compute

$$\hat{z} = arg \min_{z} \|z - z_0\|_2 + \alpha_c \mathcal{L}_{conn}(z) + \alpha_s \mathcal{L}_{stab}(z), \quad (1)$$

with α_c and α_s weighting coefficients.

Results. For each generative network G among IM-NET [3] and PQ-NET [12], we provide visual results paired with gradient visualization over the optimization steps in Figures 7 and 8 respectively.

IM-NET [3] based optimization is performed by decoding latent vectors at resolution 32 (G(z)) which are then fed to the physical modules as explained in our paper. Figure 7 shows the evolution of $G(z)$ over the optimization steps. We also display the meshes corresponding to the initial and final shapes, G(z_0) and G($\hat{z}$) respectively, decoded at resolution 256. For connectivity optimization, the loss gradient is non null only in voxels that link connected components. As for the stability optimization, We display normalized gradient values. For clarity of visualization, we set gradients in voxels that remain outside the shape surface to 0 to avoid blur. Remark that the gradient has higher negative values around missing parts (to add geometry) and higher positive values at redundant parts (to remove geometry) for reaching stability. In general, both physical losses manage to correct the initial non valid shape. We highlight that combining both losses is necessary to obtain optimal results. For instance, consider the example from the second row in Figure 7a. We remark that topology loss manages to recover shape connectivity. However, the resulting shape is connected but unstable. To see this, we expect that removing the disconnected geometry from the front leg leads to connecting the shape with minimum geometry variation, as advised by the loss function. Consequently, the need for an additional stability term to rather promote the solution based on lengthening the front leg proves necessary. This is illustrated by Figure 7c first row, where we optimize the same latent vector using both losses.

PQ-NET [12] based optimization is performed by decoding latent vectors at part resolution 32. Note that for the other experiments in the main paper we decrease the resolution to 16 for the connectivity module only, for memory cost. Similarly to the experiments above, Figure 8 shows the evolution of $G(z)$ as well as the initial and final shape meshes obtained by decoding G(z_0) and G($\hat{z}$) at resolution 256. The connectivity module is applied on

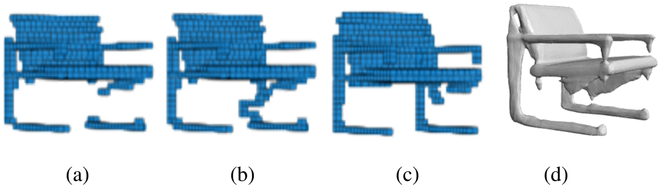


Figure 6: Visual results for 3D shape optimization based on IM-Net [3] model. (a) The input shape to optimize represented by its latent vector z_0 and the corresponding occupancy field V_0 at resolution 32. (b) Output shape achieved by geometry optimization $\hat{V} = arg\min_V \mathcal{L}_c(V) + \|V - V_0\|$. (c) and (d) Output shape achieved by latent optimization $\hat{z} = arg\min_z \mathcal{L}_c(z) + \|z - z_0\|$ decoded at resolution 32 and 256 respectively. The direct geometry optimization follows the path with minimum variations to correct the shape, while the latent optimization approach demonstrates an understanding of the shape structure and delivers plausible result.

part occupancy fields. To understand this, the shape points in Figure 8a with black contour correspond to part voxel centers that belong to shape geometry (have occupancy value bigger than 0.5). We also display the gradient values among all parts occupancy fields represented by the voxel centers colored by gradient intensity. The connectivity loss gradient is non null only in voxels that link connected components. As for the stability optimization, we feed the point cloud shape obtained as described in the main paper to the physical module. Figure 8b shows the evolution of the decoded point cloud, starting from the first shape, along with the physical stability gradient arrows. Gradient arrows are multiplied by -1 to show the displacement direction promoted by the stability loss. Remark that, in general, gradient arrows point towards enlarging the shape basis, this in line with stability intuition. Besides, since connectivity is ill-defined when working with point clouds, the obtained shapes illustrate a sufficient evolution towards stable result while suffering from connectivity failures. Consider the example from the second row in Figure 8b. The stability optimization attempts to recover the missing legs. It however misses the connectivity requirement. An optimal result is achieved when conducting the same experiment using both losses as demonstrated in the second row from Figure 8c.

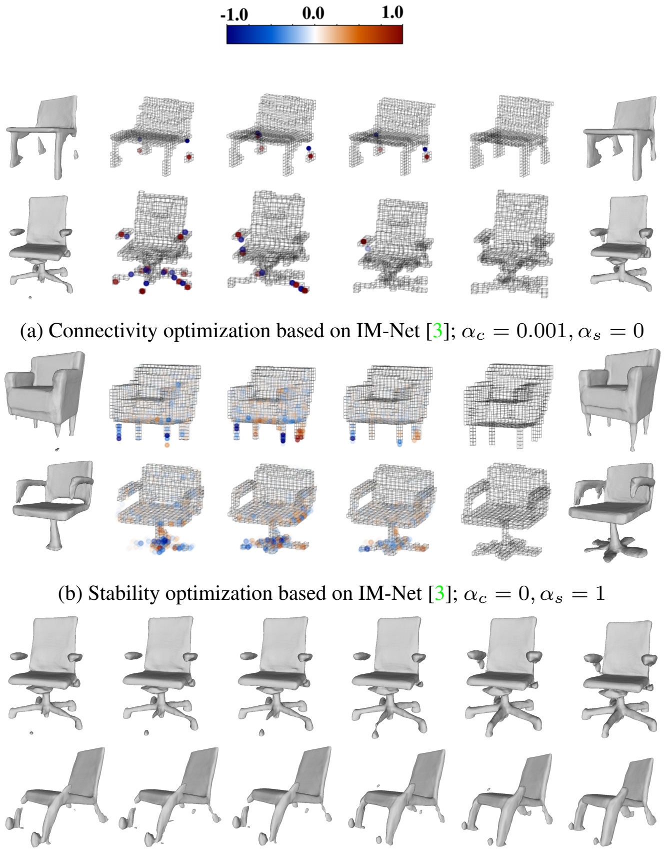
Overall, the achieved results reveal the efficiency of our physical losses, and confirm the relevance of combining generative modeling with physical reasoning to produce shapes that are both visually plausible and physically valid.

Comparison to several baselines

- In relation to the work in [6], their post-processing algorithm needs part relationship annotations including Adjacent and Support that we don't require. Besides, it only addresses connectivity via part adjacency and support via equal parts height (see Figure 7 in [6]). Our approach is significantly more generic and handles more diverse and challenging failures. Please also note that all our losses are fully differentiable and can also be used for post-processing via test-time optimization.
- In contrast to [1] that proposes a shape optimization algorithm, we consider the task of shape generation that requires particular attention to the generated shape visual quality as well. To this end, instead of directly optimizing the generated shape parameters, we rather optimize its latent representation in order to constrain the search space to the learned latent space of plausible objects. For the sake of clarity, Figure 6 compares both optimization approaches performed on example input (6a) sampled using IM-Net[3]-based generator. While the direct geometry optimization (6b) follows the path with minimum variations to correct the shape, the latent optimization (6c) approach demonstrates an understanding of the shape structure and produces more plausible result. Furthermore, the latent optimization allows to decode the optimal shape at high resolution (6d). This is not feasible when directly optimizing the shape geometry.
- Several other approaches [9, 10, 13] also focused on shape stability optimization. These approaches only consider shape deformation or shape carving techniques to stabilize the shape. These operations are not sufficient in our case. In fact, we need more complicated shape modifications such as part addition.

5. Comparison to the topological layer in [5]

Learning deep generative network using topological priors is pioneered by [5]. In relation to their work, our topological regularization differ in several key aspects. First, our definition of the *connected components* takes into account the iso-surface values between the birth and death intervals. Second, we operate in the latent space and we found that this prevents the generative model from collapsing. Finally, the implementation of [5] counts on the full (including higher-dimensional) homology and uses a simplicial complex, whereas we use the cubical complex and directly compute 0-dimensional homology only. This results in our topology loss being **225 times** faster than when using *levelset_dionysus.Diagramlayer* from [5] code (their other *LevelSetLayer* doesn't support cyclic complex), which is essential for making the network training tractable. For completeness, we ran their suggested approach on 3DGAN [11] with our experimental data and found that their finetuning leads to a connection ratio CR of $\mathbf{20.8\%}$ compared to $\mathbf{42.6\%}$, achieved by our closest architecture 3DVAE. Note



(a) Connectivity optimization based on IM-Net [3]; $\alpha_c = 0.001, \alpha_s = 0$

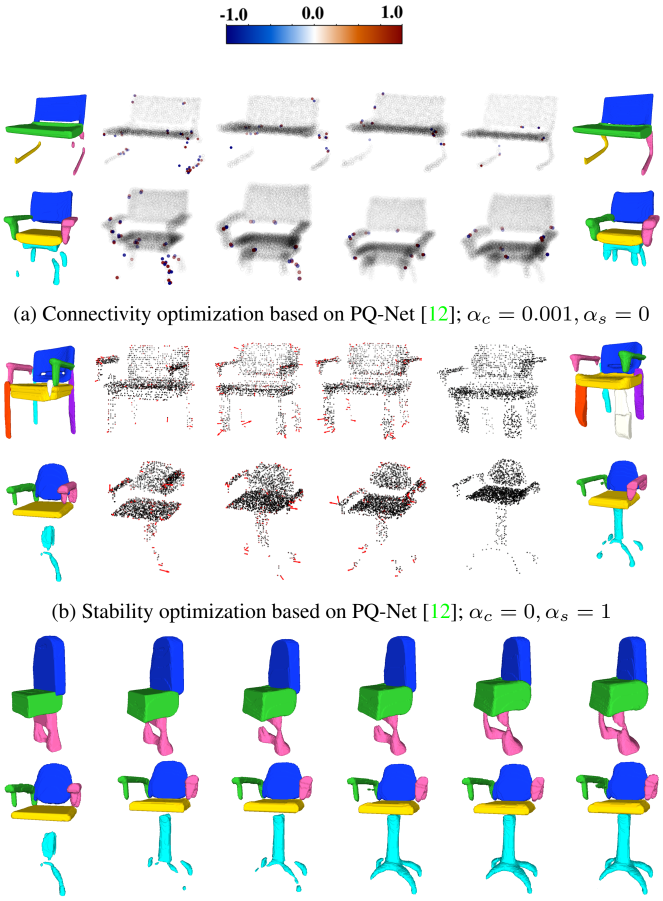
(b) Stability optimization based on IM-Net [3]; $\alpha_c = 0, \alpha_s = 1$

(c) Physical optimization based on IM-Net [3]; $\alpha_c = 0.001, \alpha_s = 1$

Figure 7: Visual results of IM-Net [3] based shape optimization. (a) (b) From left to right: initial shape at resolution 256, 4 intermediate shapes at resolution 32 paired with gradient values, the final shape at resolution 256. (c) From left to right: initial shape, 4 intermediate shapes and final shape at resolution 256. All shapes are randomly sampled from the latent space. Connectivity and stability losses deliver connected and stable shapes respectively while being visually plausible.

that their topological regularization helps reducing noise but in most cases do not connect shape parts. In fact, longer training is needed to improve connectivity. However, this leads to deviating from original shapes and to reducing shape quality. We sidestep this problem by carefully selecting the connected components to optimize based on isosurface values so that we avoid unnecessary shape variations.

Timing Our topology layer takes $0.96s$ and $0.005s$ for the forward and backward passes respectively when applied on a batch of 32 occupancy grids of resolution 32 on Intel Xeon 5220 Gold CPU.



(a) Connectivity optimization based on PQ-Net [12]; $\alpha_c = 0.001, \alpha_s = 0$

(b) Stability optimization based on PQ-Net [12]; $\alpha_c = 0, \alpha_s = 1$

(c) Physical optimization based on PQ-Net [12]; $\alpha_c = 0.001, \alpha_s = 1$

Figure 8: Visual results of PQ-Net [12] based shape optimization. (a) (b) From left to right: initial shape at part resolution 256, 4 intermediate shapes at part resolution 32 paired with gradient values, the final shape at part resolution 256. (c) From left to right: initial shape, 4 intermediate shapes and final shape at part resolution 256. All shapes are randomly sampled from the latent space. Combining connectivity and stability losses yields the best results.

References

[1] 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. 5

[2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. Technical Report 1512.03012, arXiv preprint, Dec. 2015. 3

[3] 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, 4, 5, 6

[4] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2019. 3, 4

[5] 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. 1, 5

[6] 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. 5

[7] 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. 3

[8] 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

[9] 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. 5

[10] Lingfeng Wang and Emily Whiting. Buoyancy optimization for computational fabrication. *Computer Graphics Forum (Proceedings of Eurographics)*, 35(2), 2016. 5

[11] 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. 5

[12] 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, 4, 6

[13] 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. 5