DiscoNet: Shapes Learning on Disconnected Manifolds for 3D Editing Supplementary Material

Éloi Mehr[1], Ariane Jourdan , Nicolas Thome[1], Matthieu Cord[1], and Vincent Guitteny

[1]LIP6, Sorbonne Université

1. Theoretical complements

1.1. Mathematical statements

In this subsection, we first prove Lemma 1, needed to prove then Theorem 1 appearing in our paper. Afterward, we discuss its consequences. In all the following statements, d denotes the canonical Euclidean metric, *i.e.* the metric induced by the L^2 norm, and $B(a, R)$ is the open ball of center a and radius R for the Euclidean metric d.

Lemma 1. *For $p, m \in \mathbb{N} \setminus \{0\}$, let $g_\theta \colon \mathbb{R}^p \to \mathbb{R}^m$ be a decoder,* i.e. *a neural network parameterized by θ, whose activations are either ReLU, leaky ReLU, or differentiable activations with bounded derivative, and which can also include max pooling layers. Then for any $r > 0$, there exists $k_r > 0$, such that for any weights θ with $\|\theta\|_\infty \leq r$, and any $x, y \in \mathbb{R}^p$,*

$$d(g_\theta(x), g_\theta(y)) \leq k_r\, d(x, y)\,, \tag{1}$$

Proof. Let g_θ be a decoder from $\mathbb{R}^p \to \mathbb{R}^m$, $p, m \in \mathbb{N} \setminus \{0\}$, with L layers:

$$g_\theta = g^{(1)}_{\theta_1} \circ \ldots \circ g^{(L)}_{\theta_L}\,, \tag{2}$$

where θ_i is the sub-vector of θ that represents the weights of the i-th layer. Each function g_{θ_i} is a mapping from $\mathbb{R}^{n_i}$ to $\mathbb{R}^{n_{i+1}}$, all $n_i \geq 1$. Let $r > 0$ and assume $\|\theta\|_\infty \leq r$.

The first possibility for the i-th layer is to be a max pooling layer. In that case, we can write for all $j \in [\![1, n_{i+1}]\!]$, and for all $x \in \mathbb{R}^{n_i}$:

$$\left(g^{(i)}_{\theta_i}(x)\right)_j = \max_{l \in V_j} x_l\,, \tag{3}$$

where V_j is a non-empty subset of $[\![1, n_i]\!]$. Let $y \in \mathbb{R}^{n_i}$. Then we have

$$\left|\left(g^{(i)}_{\theta_i}(x)\right)_j - \left(g^{(i)}_{\theta_i}(y)\right)_j\right| = \left|\max_{l \in V_j} x_l - \max_{l \in V_j} y_l\right| \tag{4}$$

$$\leq \max_{l \in V_j} |x_l - y_l| \tag{5}$$

$$\leq \|x - y\|_\infty \tag{6}$$

$$\leq d(x, y)\,. \tag{7}$$

where Equation (5) is a well-known inequality, that we derive for completeness in Lemma 2. So we get from Equation (7):

$$d(g^{(i)}_{\theta_i}(x), g^{(i)}_{\theta_i}(y)) \leq \sqrt{n_{i+1}}\left\|g^{(i)}_{\theta_i}(x) - g^{(i)}_{\theta_i}(y)\right\|_\infty \tag{8}$$

$$\leq k_i\, d(x, y)\,, \tag{9}$$

where $k_i = \sqrt{n_{i+1}}$.

The second possibility for the i-th layer is to be a learnable layer with an activation function denoted σ_i. In that case we decompose the weights $\theta_i = (w^{(i)}, b^{(i)})$, such that for all $j \in [\![1, n_{i+1}]\!]$, and for all $x \in \mathbb{R}^{n_i}$:

$$\left(g^{(i)}_{\theta_i}(x)\right)_j = \sigma_i\left(\left\langle w^{(i)}_j, x\right\rangle + b^{(i)}_j\right)\,, \tag{10}$$

If σ_i is differentiable with bounded derivative, the mean value inequality implies that there exists $c_i > 0$, such that for all $x, y \in \mathbb{R}$,

$$|\sigma_i(x) - \sigma_i(y)| \leq c_i\, |x - y|\,. \tag{11}$$

Besides, if σ_i is a ReLU activation or a leaky ReLU, Equation (11) also holds with $c_i = 1$.[1] Thus, we get for all

[1]Notice that the existence of a constant $c_i > 0$ for Equation (11) obviously still holds when a layer shares different activation functions.

$x, y \in \mathbb{R}^{n_i}$

$$
\begin{aligned}
d(g_{\theta_i}^{(i)}(x), g_{\theta_i}^{(i)}(y)) &\leq \left\| g_{\theta_i}^{(i)}(x) - g_{\theta_i}^{(i)}(y) \right\|_1 && (12)\\
&\leq \sum_{j=1}^{n_{i+1}} c_i \left| \left\langle w_j^{(i)}, x \right\rangle - \left\langle w_j^{(i)}, y \right\rangle \right| && (13)\\
&= c_i \sum_{j=1}^{n_{i+1}} \left| \sum_{l=1}^{n_i} w_{j,l}^{(i)} (x_l - y_l) \right| && (14)\\
&\leq c_i \sum_{j=1}^{n_{i+1}} \sum_{l=1}^{n_i} \left| w_{j,l}^{(i)} \right| |x_l - y_l| && (15)\\
&\leq c_i \|\theta\|_\infty n_{i+1} \|x - y\|_1 && (16)\\
&\leq c_i \|\theta\|_\infty n_{i+1} \sqrt{n_i}\, d(x, y) && (17)\\
&\leq k_i\, d(x, y)\,, && (18)
\end{aligned}
$$

with $k_i = c_i \times r \times n_{i+1} \times \sqrt{n_i}$.

Eventually, in both cases we have

$$d(g_{\theta_i}^{(i)}(x), g_{\theta_i}^{(i)}(y)) \leq k_i\, d(x, y)\,, \tag{19}$$

and so if we set

$$k_r = \prod_{i=1}^{l} k_i > 0\,, \tag{20}$$

we finally get for all weights θ with $\|\theta\|_\infty \leq r$ and for all $x, y \in \mathbb{R}^p$:

$$d(g_\theta(x), g_\theta(y)) \leq k_r\, d(x, y)\,. \tag{21}$$

■

For completeness we prove the following lemma used in Lemma 1.

Lemma 2. *For any $x, y \in \mathbb{R}^n$, $n \in \mathbb{N} \setminus \{0\}$, we have:*

$$\left| \max_l x_l - \max_l y_l \right| \leq \max_l |x_l - y_l| = \|x - y\|_\infty\,. \tag{22}$$

Proof. For any $l \in [\![1, n]\!]$, we have

$$x_l \leq |x_l - y_l| + y_l\,, \tag{23}$$

and so

$$
\begin{aligned}
\max_l x_l &\leq \max_l \left(|x_l - y_l| + y_l \right) && (24)\\
&\leq \max_l |x_l - y_l| + \max_l y_l\,, && (25)
\end{aligned}
$$

which implies

$$\max_l x_l - \max_l y_l \leq \max_l |x_l - y_l|\,. \tag{26}$$

Similarly, we get

$$\max_l y_l - \max_l x_l \leq \max_l |x_l - y_l|\,, \tag{27}$$

which finally allows to conclude the proof:

$$\left| \max_l x_l - \max_l y_l \right| \leq \max_l |x_l - y_l| = \|x - y\|_\infty\,. \tag{28}$$

■

We now prove our main theorem.

Theorem 1. *Let $g_\theta \colon \mathbb{R}^p \to \mathbb{R}^m$ be a decoder, $p, m \in \mathbb{N} \setminus \{0\}$. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two subsets of $\mathbb{R}^m$ such that $d(\mathcal{M}_1, \mathcal{M}_2) > 0$.*

Then for any $r > 0$, there exists $C_r > 0$, such that for any weights θ with $\|\theta\|_\infty \leq r$, and for any continuous path $\gamma \colon [0, 1] \to \mathbb{R}^p$ with $\gamma(0) \in g_\theta^{-1}(\mathcal{M}_1)$ and $\gamma(1) \in g_\theta^{-1}(\mathcal{M}_2)$, there exists $h \in \gamma(]0, 1[)$ such that:

$$g_\theta(B(h, C_r)) \subset (\mathcal{M}_1 \cup \mathcal{M}_2)^c\,. \tag{29}$$

Proof. Let g_θ be a decoder from $\mathbb{R}^p \to \mathbb{R}^m$, $p, m \in \mathbb{N} \setminus \{0\}$, $\mathcal{M}_1$ and $\mathcal{M}_2$ be two subsets of $\mathbb{R}^m$ such that $d(\mathcal{M}_1, \mathcal{M}_2) > 0$, and let $r > 0$. Let $k_r > 0$ be the constant given by Lemma 1, and define

$$C_r = \frac{d(\mathcal{M}_1, \mathcal{M}_2)}{2k_r} > 0\,. \tag{30}$$

Finally, let θ be any weights such that $\|\theta\|_\infty \leq r$, and let $\gamma : [0, 1] \to \mathbb{R}^p$ continuous with $\gamma(0) \in g_\theta^{-1}(\mathcal{M}_1)$ and $\gamma(1) \in g_\theta^{-1}(\mathcal{M}_2)$.

We define

$$D_1 = d(g_\theta^{-1}(\mathcal{M}_1), g_\theta^{-1}(\mathcal{M}_2))\,. \tag{31}$$

There exist $(a_n)_{n \in \mathbb{N}}$ a sequence of elements in $g_\theta^{-1}(\mathcal{M}_1)$, and $(b_n)_{n \in \mathbb{N}}$ a sequence of elements in $g_\theta^{-1}(\mathcal{M}_2)$, such that

$$d(a_n, b_n) \underset{n \to +\infty}{\longrightarrow} D_1\,. \tag{32}$$

From Lemma 1, we have

$$d(\mathcal{M}_1, \mathcal{M}_2) \leq d(g_\theta(a_n), g_\theta(b_n)) \leq k_r\, d(a_n, b_n)\,. \tag{33}$$

Taking the limit when $n \to +\infty$, we get

$$d(\mathcal{M}_1, \mathcal{M}_2) \leq k_r\, D_1\,. \tag{34}$$

We now define

$$
\begin{aligned}
\Gamma \colon [0, 1] &\to \mathbb{R} && (35)\\
t &\mapsto d(\gamma(t), g_\theta^{-1}(\mathcal{M}_1)) - d(\gamma(t), g_\theta^{-1}(\mathcal{M}_2))\,.
\end{aligned}
$$

Γ is continuous, $\Gamma(0) \leq -D_1$, $\Gamma(1) \geq D_1$, and since $D_1 > 0$ as shown by Equation (34), there exists $t \in]0, 1[$ such that $\Gamma(t) = 0$. If we set $h = \gamma(t) \in \gamma(]0,1[)$ and $D_2 = d(h, g_\theta^{-1}(\mathcal{M}_1))$, we have

$$D_2 = d(h, g_\theta^{-1}(\mathcal{M}_1)) = d(h, g_\theta^{-1}(\mathcal{M}_2)) . \quad (36)$$

Thus, there exist $(c_n)_{n\in\mathbb{N}}$ a sequence of elements in $g_\theta^{-1}(\mathcal{M}_1)$, and $(d_n)_{n\in\mathbb{N}}$ a sequence of elements in $g_\theta^{-1}(\mathcal{M}_2)$, such that

$$d(h, c_n) \underset{n\to+\infty}{\longrightarrow} D_2 \quad \text{and} \quad d(h, d_n) \underset{n\to+\infty}{\longrightarrow} D_2 . \quad (37)$$

The triangular inequality then implies that

$$D_1 \leq d(c_n, d_n) \leq d(h, c_n) + d(h, d_n) , \quad (38)$$

and taking the limit when $n \to +\infty$ gives

$$D_1 \leq 2D_2 . \quad (39)$$

Finally, let $x \in g_\theta(B(h, C_r))$, *i.e.* $x = g_\theta(h')$ with $h' \in B(h, C_r)$. Since Equations (34) and (39) imply that

$$C_r \leq D_2 , \quad (40)$$

we get that $h' \in B(h, D_2)$. Assume that $x \in \mathcal{M}_1$, then we have

$$D_2 = d(h, g_\theta^{-1}(\mathcal{M}_1)) \leq d(h, h') < D_2 , \quad (41)$$

which is impossible. Similarly, $x \notin \mathcal{M}_2$, and we have therefore proved our claim:

$$g_\theta(B(h, C_r)) \subset (\mathcal{M}_1 \cup \mathcal{M}_2)^c . \quad (42)$$

■

We end our mathematical statements with a direct corollary of Theorem 1.

Corollary 1. *Let g_θ be a decoder from $\mathbb{R}^p \to \mathbb{R}^m$, $p, m \in \mathbb{N} \setminus \{0\}$. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two subsets of $\mathbb{R}^m$ such that $d(\mathcal{M}_1, \mathcal{M}_2) > 0$.*

Then for any $r > 0$, there exists $C_r > 0$, such that for any weights θ with $\|\theta\|_\infty \leq r$, and for any continuous path $\gamma\colon [0,1] \to \mathbb{R}^p$ with $\gamma(0) \in g_\theta^{-1}(\mathcal{M}_1)$ and $\gamma(1) \in g_\theta^{-1}(\mathcal{M}_2)$, there is a path-connected subset I of $\gamma(]0,1[)$ such that:

$$g_\theta(I) \subset (\mathcal{M}_1 \cup \mathcal{M}_2)^c \quad \text{and} \quad L(I) \geq 2C_r , \quad (43)$$

where L denotes the arc length.

Proof. Let $C_r > 0$ and $h = \gamma(t), t \in]0,1[$, as provided by the Theorem 1. Thus $\gamma(0) \notin B(h, C_r)$, and so we can define

$$t_1 = \sup \{t' \in [0,t] \mid \gamma(t') \notin B(h, C_r)\} , \quad (44)$$

and similarly

$$t_2 = \inf \{t' \in [t,1] \mid \gamma(t') \notin B(h, C_r)\} . \quad (45)$$

Then $\gamma(]t_1, t_2[) \subset B(h, C_r)$, and $\gamma(t_1), \gamma(t_2) \in \partial B(h, C_r)$ by continuity of $t' \mapsto d(\gamma(t'), h)$. By definition of the arc length

$$L(\gamma\big|_{]t_1,t_2[}) \geq L(]\gamma(t_1), \gamma(t)]) + L([\gamma(t), \gamma(t_2)[) \quad (46)$$

$$= C_r + C_r = 2C_r . \quad (47)$$

Besides, $I = \gamma(]t_1, t_2[)$ is path-connected, and by the Theorem 1, for any θ with $\|\theta\|_\infty \leq r$, we have

$$g_\theta(I) \subset (\mathcal{M}_1 \cup \mathcal{M}_2)^c \quad (48)$$

since $I \subset B(h, C_r)$. ■

We now discuss some implications of our theorem. First of all, we point out that all usual activation functions used in deep learing are either differentiable with bounded derivative (linear unit, sigmoid, tanh, arctan, sin, inverse square root unit, exponential linear unit, *etc*.), ReLU units, or leaky ReLU units. Moreover, our assumptions also include ResNet [3] architectures, as we can always represent a residual layer sequentially with an additional linear layer to duplicate the input, and a further additional linear layer to add the duplicated input to the output. Thus our assumptions on the decoder's architecture are consistent with real architectures. Besides, if necessary, we can always remove the bounded derivative assumption at the cost of bounding the latent space (*e.g*. replacing $\mathbb{R}^p$ with $[0,1]^p$) and also imposing C^1 activations, the derivative being then continuous and therefore bounded on any compact.

Let $(\mathcal{M}_1, \mathcal{M}_2)$ be a partition of the subset of plausible shapes of the input space. We assume that $d(\mathcal{M}_1, \mathcal{M}_2) > 0$, which is typically the case when $\mathcal{M}_1$ and $\mathcal{M}_2$ represent two different kinds of shapes which lie on two separated components. The first implication of the Theorem 1 comes from the following remark. As it is true for all bounded weights θ, it is especially valid for a specific θ with the constant $C_r = C_{\|\theta\|_\infty}$. The theorem then proves that it is impossible to find a (continuous) interpolating path in the latent space between any model from $\mathcal{M}_1$ to any model in $\mathcal{M}_2$, without generating implausible models, *i.e*. models that do not belong to $\mathcal{M}_1$ or $\mathcal{M}_2$. In particular, Corollary 1 of our theorem shows that any such path has a connected restriction of length at least $2C_r$ on which any model is implausible.

The second implication is that on any interpolating path in the latent space between $\mathcal{M}_1$ and $\mathcal{M}_2$, not only are some implausible models synthesized, but these models are "widely" implausible, in the sense that their latent vectors are far from both $g_\theta^{-1}(\mathcal{M}_1)$ and $g_\theta^{-1}(\mathcal{M}_2)$, as the whole ball $B(h, C_r)$ generates implausible shapes.

The last and strongest implication we draw from our theorem is that it is impossible to find a learning algorithm with only one decoder (or generator) that would bring $\mathcal{M}_1$ and $\mathcal{M}_2$ arbitrary close in the latent space. Indeed, C_r

is a constant dependent only on the distance between $\mathcal{M}_1$ and $\mathcal{M}_2$, on r, and on the architecture but not its weights as long as they are bounded by r. Thus, the only way to reduce C_r is to increase r, *i.e.* to allow the weights to grow larger in order to increase the derivatives of the network. In other words, to bring closer $\mathcal{M}_1$ and $\mathcal{M}_2$ the decoder's weights should diverge. Thus, the Theorem 1 effectively justifies our approach and the approach described in [5], which, although different, both introduce several decoders to learn a disconnected manifold.

1.2. Reassignment algorithm

We provide the pseudo code for our reassignment algorithm in the Algorithm 1. This algorithm guarantees that each autoencoder has at least $n = \lfloor \eta N \rfloor$ inputs assigned to it, thanks to the list V which stores the reassigned inputs.

Algorithm 1 Reassignment algorithm for disconnected manifold learning

Input: The mini-batch $x_1, \ldots, x_N$, the k autoencoders $g_1 \circ f_1, \ldots, g_k \circ f_k$, the minimal ratio η of shapes assigned to each autoencoder ($\eta < \frac{1}{k}$).
Output: The loss of the minibatch after reassignment.

```
Initialization
n ← ⌊ηN⌋.  ▷ The minimal number of inputs assigned to each autoencoder.
for j ∈ ⟦1, N⟧ do
    d_i^j ← d_CH(x_j, g_i ∘ f_i(x_j)).
    L_j ← min_{i∈⟦1,k⟧} d_i^j.
end for
V ← [].   ▷ V is the list of already re-assigned inputs.
Reassignment
for i ∈ ⟦1, k⟧ do
    for r ∈ ⟦1, n⟧ do ▷ r is not used, it is just a counter.
        a ← arg min_{j∈⟦1,N⟧\V} d_i^j − L_j.
        L_a ← d_i^a.
        V ← V + [a].  ▷ + denotes concatenation here.
    end for
end for
return (1/N) Σ_{j=1}^N L_j.
```

2. Implementation details

2.1. Learning

Figure 1 presents the architecture used in our experiments. All activation functions are ReLU except for the last fully connected layer (before the "add" layer) which uses tanh. We use batch normalization layers as introduced in [4]. The global max pooling layer is the usual aggregation layer used in PointNet-like architectures [7] to design an encoder acting on point clouds, just like the convolution 1d layers are the usual fully connected layers which are shared among all the points of their input in PointNet (*i.e.* shared accross their first dimension). The local max pooling layer is the graph layer used in [9] to locally aggregate features around the neighborhood of each point. All models and baselines use this architecture, except for AtlasNet. The only difference between AtlasNet and this architecture is that AtlasNet does not use the input covariances, or the local max pooling layers. For FN/AN and AtasNet, the template coordinates are taken from a unit sphere.

	chairs	*cars*	*planes*
Original AtlasNet [2]	1.81	1.75	0.98
Our AtlasNet	1.63	1.23	0.73

Table 1: Chamfer loss over the test set ($\times 10^3$).

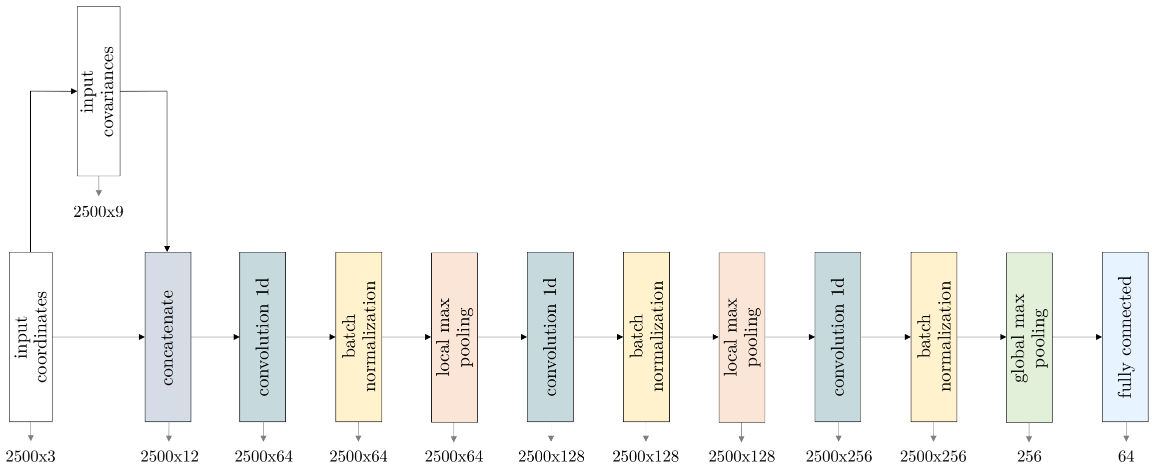
The dataset is centered, normalized into a unit sphere[2], and uniformly resampled by ray shooting the meshes, following the pre-processing step described in [8]. The resulting point clouds are uniformly subsampled to get 2500 points by model. In our learning experiments, we set $\eta = 0.25$. The training is done by mini-batch gradient descent using the ADAM optimizer [6], with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$. We run 800 epochs each time, which we have checked to be enough to reach convergence. We use a learning rate of 10^{-3} for the 700 first epochs, and 10^{-4} for the 100 last.

As AtlasNet [2] provides test errors on ShapeNet [1], with the same Chamfer loss that we use for training, we can check that our AtlasNet implementation is valid. The Table 1 compares the Chamfer loss between our own implementation of AtlasNet and the losses reported by AtlasNet original publication, using the sphere topology as input to the decoder. It validates our implementation as we get even better results than theirs, mainly because they train all ShapeNet categories together, while we train each category separately.

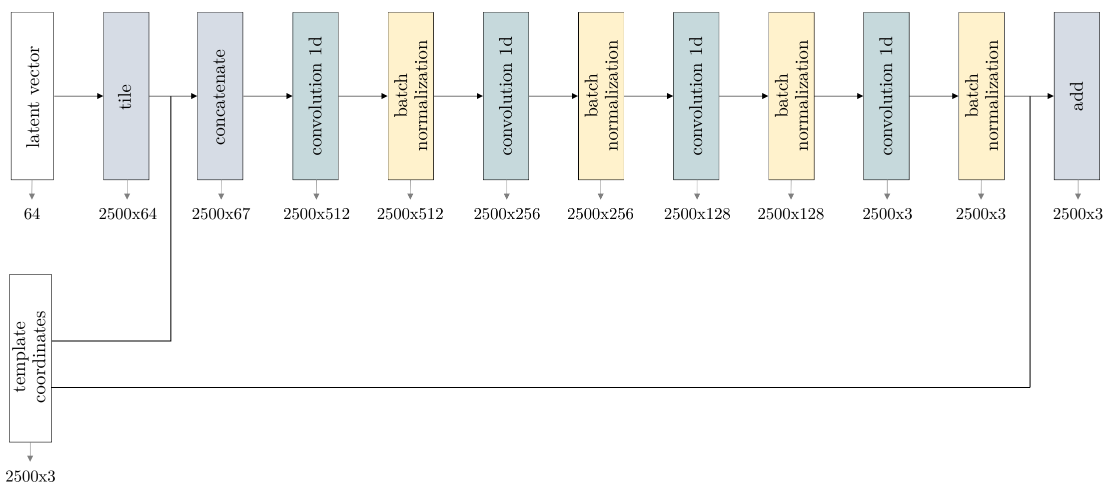
2.2. Editing

To compute the deformation field δ for the handles-based editing interface, we use the inverse quadric RBF $f(r) = 1/(1 + ar^2)$ with $a = 4$. In the 3D editing optimization, we use as threshold $\rho = 0.2$ for the energy E_c^1, and for the energy E_p we use 50 components in the Gaussian mixture model. The result of the optimization is not very sensitive to λ_s and λ_p, as long as E_c, E_s and E_p are in the same range. To compute σ in the final retargeting, we use a Gaussian kernel $K(r) = e^{-r^2/(2h)}$ with $h = 0.25$ for chairs and planes, and $h = 0.15$ for cars to better highlight the large deformations that we apply. We also keep the wheels of the edited cars fixed, by applying the retargeting only to the cars' bodies.

[2]The model is rescaled such that the furthest point to the origin lies onto the unit sphere.



(a) Encoder



(b) Decoder

Figure 1: Base 3D autoencoder architecture used in our experiments.

3. Additional results

3.1. Learning

Figures 2, 3 and 4 provide additional visual reconstruction results to illustrate the superior topology of the models reconstructed by our learning model DiscoNet ($k = 2$). The colors are transferred from the unit sphere for FN/AN and from our pre-learned templates for DiscoNet to highlight the topology and the correspondences between the vertices.

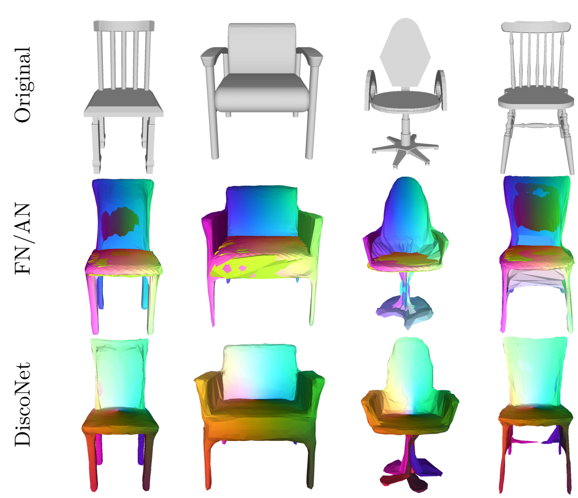


Figure 2: Reconstruction results on chairs.

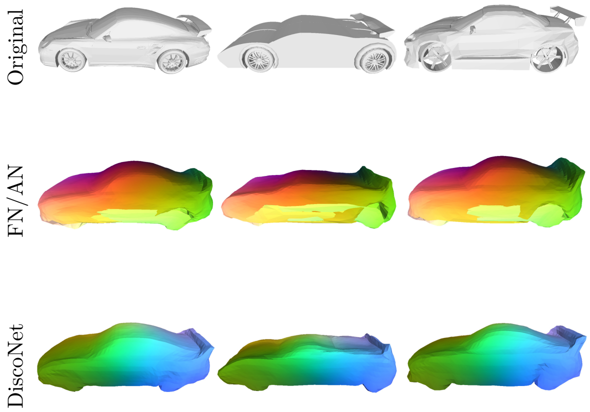


Figure 3: Reconstruction results on cars.

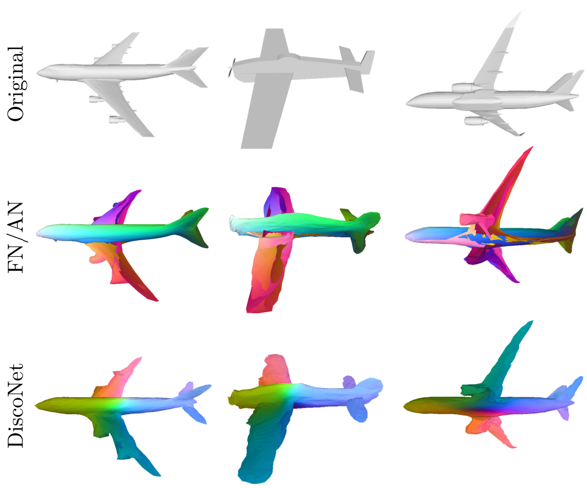


Figure 4: Reconstruction results on planes.

3.2. Editing

Figures 5 and 6 show additional editing results with our pipeline, using either the 3D handles or the 2D sketch interface. Figure 7 shows that our editing system can also be used sequentially, by iteratively optimizing the last edited model (starting from the last optimized latent vector) for each new editing.

Finally, Figure 8 presents some failure cases of our editing system, limited by our simple sketch to vertices correspondence scheme. This could probably be solved using more advanced matching algorithms, based on the arc length parameterizations of the contour and the sketch for example.

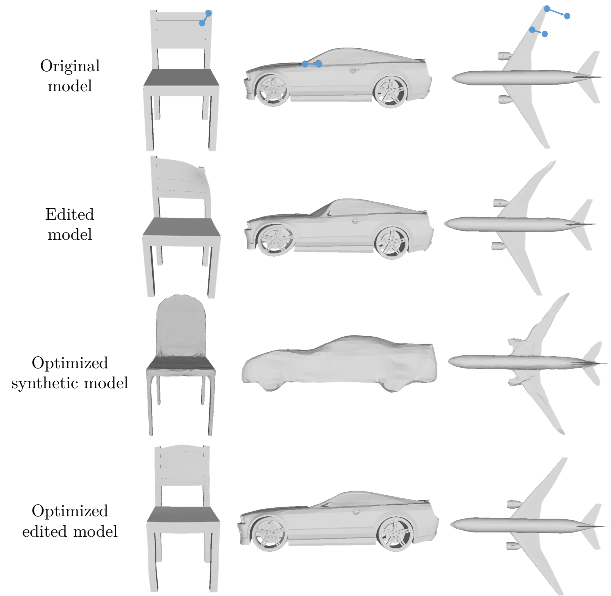


Figure 5: Results of handles-based editing.

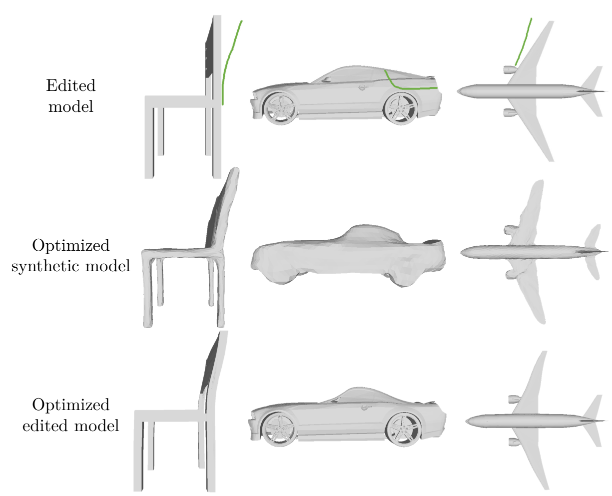


Figure 6: Results of sketch-based editing.

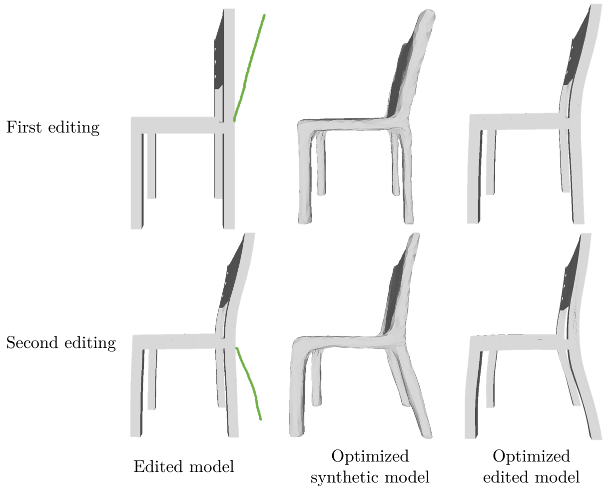


Figure 7: Result on a chair of two sequential editings.

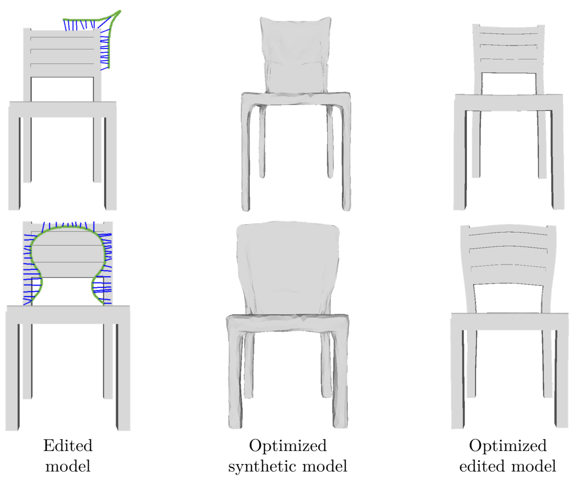


Figure 8: Illustration of two failures cases in our editing pipeline, due to a limitation in our simple sketch/contour correspondence scheme.

References

[1] 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. *arXiv preprint arXiv:1512.03012*, 2015.

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

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

[5] Mahyar Khayatkhoei, Maneesh K. Singh, and Ahmed Elgammal. Disconnected manifold learning for generative adversarial networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[7] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Conference on Computer Vision and Pattern Regognition (CVPR)*, 2017.

[8] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. In *SIGGRAPH*, 2017.

[9] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Conference on Computer Vision and Pattern Regognition (CVPR)*, 2018.