SAL: Sign Agnostic Learning of Shapes from Raw Data

Matan Atzmon and Yaron Lipman
Weizmann Institute of Science
Rehovot, Israel

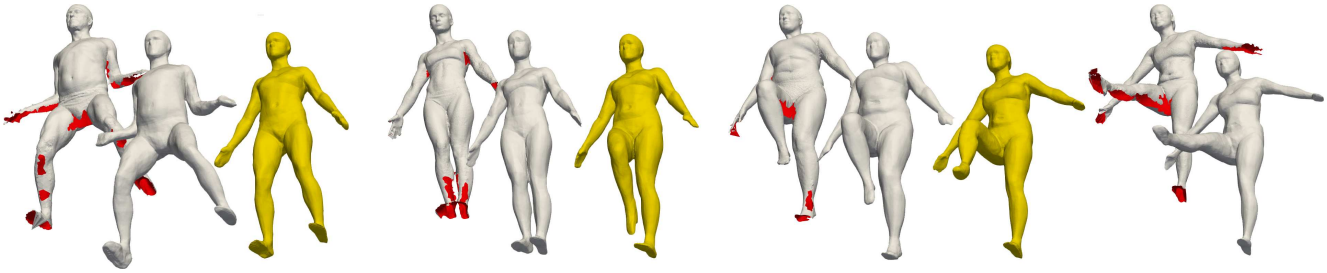


Figure 1: We introduce SAL: Sign Agnostic Learning for learning shapes directly from raw data, such as triangle soups (left in each gray pair; back-faces are in red). Right in each gray pair - the surface reconstruction by SAL of test raw scans; in gold - SAL latent space interpolation between adjacent gray shapes. Raw scans are from the D-Faust dataset [7].

Abstract

Recently, neural networks have been used as implicit representations for surface reconstruction, modelling, learning, and generation. So far, training neural networks to be implicit representations of surfaces required training data sampled from a ground-truth signed implicit functions such as signed distance or occupancy functions, which are notoriously hard to compute.

In this paper we introduce Sign Agnostic Learning (SAL), a deep learning approach for learning implicit shape representations directly from raw, unsigned geometric data, such as point clouds and triangle soups.

We have tested SAL on the challenging problem of surface reconstruction from an un-oriented point cloud, as well as end-to-end human shape space learning directly from raw scans dataset, and achieved state of the art reconstructions compared to current approaches. We believe SAL opens the door to many geometric deep learning applications with real-world data, alleviating the usual painstaking, often manual pre-process.

1. Introduction

Recently, deep neural networks have been used to reconstruct, learn and generate 3D surfaces. There are two main approaches: parametric [19, 4, 40, 15] and implicit [12, 30, 28, 2, 14, 17]. In the parametric approach neural nets are used as parameterization mappings, while the implicit approach represents surfaces as zero level-sets of neural networks:

$$\mathcal{S} = \left\{ \boldsymbol{x} \in \mathbb{R}^3 \mid f(\boldsymbol{x}; \boldsymbol{\theta}) = 0 \right\}, \tag{1}$$

where $f : \mathbb{R}^3 \times \mathbb{R}^m \to \mathbb{R}$ is a neural network, *e.g.*, multilayer perceptron (MLP). The benefit in using neural networks as implicit representations to surfaces stems from their flexibility and approximation power (*e.g.*, Theorem 1 in [2]) as well as their efficient optimization and generalization properties.

So far, neural implicit surface representations were mostly learned using a regression-type loss, requiring data samples from a ground-truth implicit representation of the surface, such as a signed distance function [30] or an occupancy function [12, 28]. Unfortunately, for the common raw form of acquired 3D data $\mathcal{X} \subset \mathbb{R}^3$, *i.e.*, a point cloud or a triangle soup[1], no such data is readily available and computing an implicit ground-truth representation for the underlying surface is a notoriously difficult task [5].

In this paper we advocate *Sign Agnostic Learning* (SAL), defined by a family of loss functions that can be used directly with raw (unsigned) geometric data $\mathcal{X}$ and produce *signed* implicit representations of surfaces. An important application for SAL is in generative models such as variational auto-encoders [24], learning shape spaces directly

[1] A triangle soup is a collection of triangles in space, not necessarily consistently oriented or a manifold.

from the raw 3D data. Figure 1 depicts an example where collectively learning a dataset of raw human scans using SAL overcomes many imperfections and artifacts in the data (left in every gray pair) and provides high quality surface reconstructions (right in every gray pair) and shape space (interpolations of latent representations are in gold).

We have experimented with SAL for surface reconstruction from point clouds as well as learning a human shape space from the raw scans of the D-Faust dataset [7]. Comparing our results to current approaches and baselines we found SAL to be the method of choice for learning shapes from raw data, and believe SAL could facilitate many computer vision and computer graphics shape learning applications, allowing the user to avoid the tedious and unsolved problem of surface reconstruction in preprocess.

2. Previous work

2.1. Surface learning with neural networks

Neural parameteric surfaces. One approach to represent surfaces using neural networks is parametric, namely, as parameterization charts $f : \mathbb{R}^2 \to \mathbb{R}^3$. Groueix et al. [19] suggest to represent a surface using a collection of such parameterization charts (*i.e.*, atlas); Williams et al. [40] optimize an atlas with proper transition functions between charts and concentrate on reconstructions of individual surfaces. Sinha et al. [32, 33] use geometry images as global parameterizations, while [27] use conformal global parameterizations to reduce the number of degrees of freedom of the map. Parametric representation are explicit but require handling of coverage, overlap and distortion of charts.

Neural implicit surfaces. Another approach to represent surfaces using neural networks, which is also the approach taken in this paper, is using an implicit representation, namely $f : \mathbb{R}^3 \to \mathbb{R}$ and the surface is defined as its zero level-set, equation 1. Some works encode f on a volumetric grid such as voxel grid [41] or an octree [36]. More flexibility and potentially more efficient use of the degrees of freedom of the model are achieved when the implicit function f is represented as a neural network [12, 30, 28, 2, 17]. In these works the implicit is trained using a regression loss of the signed distance function [30], an occupancy function [12, 28] or via particle methods to directly control the neural level-sets [2]. Excluding the latter that requires sampling the zero level-set, all regression-based methods require ground-truth inside/outside information to train the implicit f. In this paper we present a sign agnostic training method, namely training method that can work directly with the raw (unsigned) data.

Shape representation learning. Learning collections of shapes is done using Generative Adversarial Networks (GANs) [18], auto-encoders and variational auto-encoders [24], and auto-decoders [8]. Wu et al. [41] use GAN on a voxel grid encoding of the shape, while Ben-Hamu et al. [4] apply GAN on a collection of conformal charts. Dai et al. [13] use encoder-decoder architecture to learn a signed distance function to a complete shape from a partial input on a volumetric grid. Stutz et al. [34] use variational auto-encoder to learn an implicit surface representations of cars using a volumetric grid. Baqautdinov et al. [3] use variational auto-encoder with a constant mesh to learn parametrizations of faces shape space. Litany et al. [25] use variational auto-encoder to learn body shape embeddings of a template mesh. Park et al. [30] use auto-decoder to learn implicit neural representations of shapes, namely directly learns a latent vector for every shape in the dataset. In our work we also make use of a variational auto-encoder but differently from previous work, learning is done directly from raw 3D data.

2.2. Surface reconstruction.

Signed surface reconstruction. Many surface reconstruction methods require normal or inside/outside information. Carr et al. [9] were among the first to suggest using a parametric model to reconstruct a surface by computing its implicit representation; they use radial basis functions (RBFs) and regress at inside and outside points computed using oriented normal information. Kazhdan et al. [22, 23] solve a Poisson equation on a volumetric discretization to extend points and normals information to an occupancy indicator function. Walder et al. [38] use radial basis functions and solve a variational hermite problem (*i.e.*, fitting gradients of the implicit to the normal data) to avoid trivial solution. In general our method works with a non-linear parameteric model (MLP) and therefore does not require a-priori space discretization nor works with a fixed linear basis such as RBFs.

Unsigned surface reconstruction. More related to this paper are surface reconstruction methods that work with unsigned data such as point clouds and triangle soups. Zhao et al. [43] use the level-set method to fit an implicit surface to an unoriented point cloud by minimizing a loss penalizing distance of the surface to the point cloud achieving a sort of minimal area surface interpolating the points. Walder et al. [37] formulates a variational problem fitting an implicit RBF to an unoriented point cloud data while minimizing a regularization term and maximizing the norm of the gradients; solving the variational problem is equivalent to an eigenvector problem. Mullen et al. [29] suggests to sign an unsigned distance function to a point cloud by a multi-stage algorithm first dividing the problem to near and far

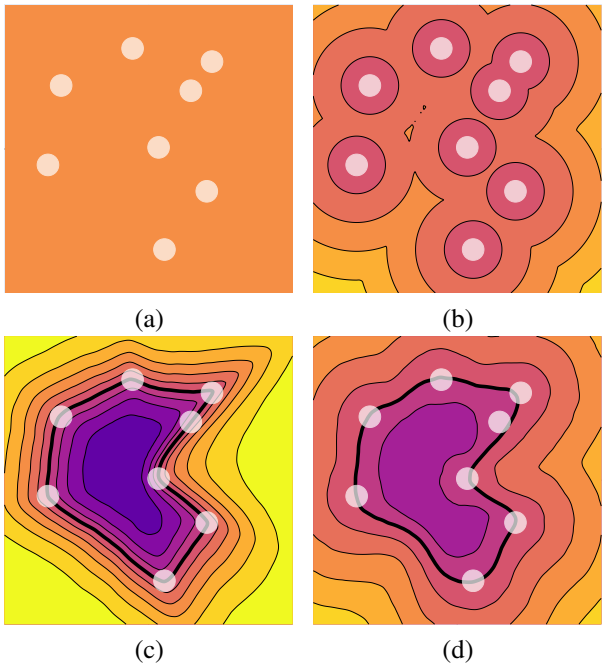


Figure 2: Experiment with sign agnostic learning in 2D: (a) and (b) show the unsigned L^0 and L^2 (resp.) distances to a 2D point cloud (in gray); (c) and (d) visualize the different level-sets of the neural networks optimized with the respective sign agnostic losses. Note how the zero level-sets (in bold) gracefully connect the points to complete the shape.

field sign estimation, and propagating far field estimation closer to the zero level-set; then optimize a convex energy fitting a smooth sign function to the estimated sign function. Takayama et al. [35] suggested to orient triangle soups by minimizing the Dirichlet energy of the generalized winding number noting that correct orientation yields piecewise constant winding number. Xu et al. [42] suggested to compute robust signed distance function to triangle soups by using an offset surface defined by the unsigned distance function. Zhiyang et al. [21] fit an RBF implicit by optimizing a non-convex variational problem minimizing smoothness term, interpolation term and unit gradient at data points term. All these methods use some linear function space; when the function space is global, *e.g.* when using RBFs, model fitting and evaluation are costly and limit the size of point clouds that can be handled efficiently, while local support basis functions usually suffer from inferior smoothness properties [39]. In contrast we use a non-linear function basis (MLP) and advocate a novel and simple sign agnostic loss to optimize it. Evaluating the non-linear neural network model is efficient and scalable and the training process can be performed on a large number of points, *e.g.*, with stochastic optimization techniques.

3. Sign agnostic learning

Given a raw input geometric data, $\mathcal{X} \subset \mathbb{R}^3$, *e.g.*, a point cloud or a triangle soup, we are looking to optimize the weights $\boldsymbol{\theta} \in \mathbb{R}^m$ of a network $f(\boldsymbol{x};\boldsymbol{\theta})$, where $f : \mathbb{R}^3 \times \mathbb{R}^m \to \mathbb{R}$, so that its zero level-set, equation 1, is a surface approximating $\mathcal{X}$.

We introduce the *Sign Agnostic Learning* (SAL) defined by a loss of the form

$$\text{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}\sim D_{\mathcal{X}}}\, \tau\Big(f(\boldsymbol{x};\boldsymbol{\theta}), h_{\mathcal{X}}(\boldsymbol{x})\Big), \tag{2}$$

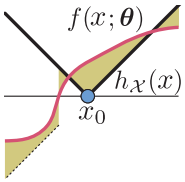
where $D_{\mathcal{X}}$ is a probability distribution defined by the input data $\mathcal{X}$; $h_{\mathcal{X}}(\boldsymbol{x})$ is some *unsigned* distance measure to $\mathcal{X}$; and $\tau : \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}$ is a differentiable *unsigned similarity function* defined by the following properties:

(i) *Sign agnostic:* $\tau(-a, b) = \tau(a, b), \forall a \in \mathbb{R}, b \in \mathbb{R}_+$.

(ii) *Monotonic:* $\frac{\partial \tau}{\partial a}(a, b) = \rho(a - b), \forall a, b \in \mathbb{R}_+$,

where $\rho : \mathbb{R} \to \mathbb{R}$ is a monotonically increasing function with $\rho(0) = 0$. An example of an unsigned similarity is $\tau(a, b) = ||a| - b|$.

To understand the idea behind the definition of the SAL loss, consider first a standard regression loss using $\tau(a, b) = |a - b|$ in equation 2. This would encourage f to resemble the unsigned distance $h_{\mathcal{X}}$ as much as possible. On the other hand, using the unsigned similarity τ in equation 2 introduces a new local minimum of loss where f is a *signed* function such that $|f|$ approximates $h_{\mathcal{X}}$. To get this desirable local minimum we later design a network weights' initialization $\boldsymbol{\theta}^0$ that favors the signed local minima.



As an illustrative example, the inset depicts the one dimensional case ($d = 1$) where $\mathcal{X} = \{x_0\}$, $h_{\mathcal{X}}(x) = |x - x_0|$, and $\tau(a, b) = ||a| - b|$, which satisfies properties (i) and (ii), as discussed below; the loss therefore strives to minimize the area of the yellow set. When initializing the network parameters $\boldsymbol{\theta} = \boldsymbol{\theta}^0$ properly, the minimizer $\boldsymbol{\theta}^*$ of loss defines an implicit $f(x;\boldsymbol{\theta}^*)$ that realizes a *signed* version of $h_{\mathcal{X}}$; in this case $f(x;\boldsymbol{\theta}^*) = x - x_0$. In the three dimensional case the zero level-set $\mathcal{S}$ of $f(\boldsymbol{x};\boldsymbol{\theta}^*)$ will represent a surface approximating $\mathcal{X}$.

To theoretically motivate the loss family in equation 2 we will prove that it possess a *plane reproduction* property. That is, if the data $\mathcal{X}$ is contained in a plane, there is a critical weight $\boldsymbol{\theta}^*$ reconstructing this plane as the zero level-set of $f(\boldsymbol{x};\boldsymbol{\theta}^*)$. Plane reproduction is important for surface approximation since surfaces, by definition, have an approximate tangent plane almost everywhere [16].

We will explore instantiations of SAL based on different choices of unsigned distance functions $h_{\mathcal{X}}$, as follows.

Unsigned distance functions. We consider two p-distance functions: For $p = 2$ we have the standard L^2 (Euclidean) distance

$$h_2(\boldsymbol{z}) = \min_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{z} - \boldsymbol{x}\|_2 \,, \tag{3}$$

and for $p = 0$ the L^0 distance

$$h_0(\boldsymbol{z}) = \begin{cases} 0 & \boldsymbol{z} \in \mathcal{X} \\ 1 & \boldsymbol{z} \notin \mathcal{X} \end{cases} . \tag{4}$$

Unsigned similarity function. Although many choices exist for the unsigned similarity function, in this paper we take

$$\tau_\ell(a, b) = \big||a| - b\big|^\ell \,, \tag{5}$$

where $\ell \geq 1$. The function τ_ℓ is indeed an unsigned similarity: it satisfies (i) due to the symmetry of $|\cdot|$; and since $\frac{\partial \tau}{\partial a} = \ell \big||a| - b\big|^{\ell-1} \operatorname{sign}(a - b\operatorname{sign}(a))$ it satisfies (ii) as-well.

Distribution $D_{\mathcal{X}}$. The choice of $D_{\mathcal{X}}$ is depending on the particular choice of $h_{\mathcal{X}}$. For L^2 distance, it is enough to make the simple choice of splatting an isotropic Gaussian, $\mathcal{N}(\boldsymbol{x}, \sigma^2 I)$, at every point (uniformly randomized) $\boldsymbol{x} \in \mathcal{X}$; we denote this probability $\mathcal{N}_\sigma(\mathcal{X})$; note that σ can be taken to be a function of $\boldsymbol{x} \in \mathcal{X}$ to reflect local density in $\mathcal{X}$. In this case, the loss takes the form

$$\operatorname{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}_\sigma(\mathcal{X})} \big||f(\boldsymbol{z};\boldsymbol{\theta})| - h_2(\boldsymbol{z})\big|^\ell . \tag{6}$$

For the L^0 distance however, $h_{\mathcal{X}}(\boldsymbol{x}) \neq 1$ only for $\boldsymbol{x} \in \mathcal{X}$ and therefore a non-continuous density should be used; we opt for $\mathcal{N}(\boldsymbol{x}, \sigma^2 I) + \delta_{\boldsymbol{x}}$, where $\delta_{\boldsymbol{x}}$ is the delta distribution measure concentrated at $\boldsymbol{x}$. The loss takes the form

$$\operatorname{loss}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}_\sigma(\mathcal{X})} \big||f(\boldsymbol{z};\boldsymbol{\theta})| - 1\big|^\ell + \mathbb{E}_{\boldsymbol{x}\sim\mathcal{X}} \big|f(\boldsymbol{x};\boldsymbol{\theta})\big|^\ell . \tag{7}$$

Remarkably, the latter loss requires only randomizing points $\boldsymbol{z}$ near the data samples without any further computations involving $\mathcal{X}$. This allows processing of large and/or complex geometric data.

Neural architecture. Although SAL can work with different parametric models, in this paper we consider a multilayer perceptron (MLP) defined by

$$f(\boldsymbol{x};\boldsymbol{\theta}) = \varphi\big(\boldsymbol{w}^T f_\ell \circ f_{\ell-1} \circ \cdots \circ f_1(\boldsymbol{x}) + b\big), \tag{8}$$

and

$$f_i(\boldsymbol{y}) = \nu(\boldsymbol{W}_i \boldsymbol{y} + \boldsymbol{b}_i), \boldsymbol{W} \in \mathbb{R}^{d_i^{out} \times d_i^{in}}, \boldsymbol{b}_i \in \mathbb{R}^{d_i^{out}}, \tag{9}$$

where $\nu(a) = (a)_+$ is the ReLU activation, and $\boldsymbol{\theta} = (\boldsymbol{w}, b, \boldsymbol{W}_\ell, \boldsymbol{b}_\ell, \ldots, \boldsymbol{W}_1, \boldsymbol{b}_1)$; φ is a strong non-linearity, as defined next:

(a) (b) (c)

Figure 3: Geometric initialization of neural networks: An MLP with our weight initialization (see Theorem 1) is approximating the signed distance function to an r-radius sphere, $f(\boldsymbol{x};\boldsymbol{\theta}^0) \approx \varphi(\|\boldsymbol{x}\| - r)$, where the approximation improves with the width of the hidden layers: (a) depicts an MLP with 100-neuron hidden layers; (b) with 200; and (c) with 2000.

Definition 1. *The function $\varphi : \mathbb{R} \to \mathbb{R}$ is called a* strong non-linearity *if it is differentiable (almost everywhere), anti-symmetric, $\varphi(-a) = -\varphi(a)$, and there exists $\beta \in \mathbb{R}_+$ so that $\beta^{-1} \geq \varphi'(a) \geq \beta > 0$, for all $a \in \mathbb{R}$ where it is defined.*

In this paper we use $\varphi(a) = a$ or $\varphi(a) = \tanh(a) + \gamma a$, where $\gamma \geq 0$ is a parameter. Furthermore, similarly to previous work [30, 12] we have incorporated a skip connection layer s, concatenating the input $\boldsymbol{x}$ to the middle hidden layer, that is $s(\boldsymbol{y}) = (\boldsymbol{y}, \boldsymbol{x})$, where here $\boldsymbol{y}$ is a hidden variable in f.

2D example. The two examples in Figure 2 show case the SAL for a 2D point cloud, $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^8 \subset \mathbb{R}^2$, (shown in gray) as input. These examples were computed by optimizing equation 6 (right column) and equation 7 (left column) with $\ell = 1$ using the L^2 and L^0 distances (resp.). The architecture used is an 8-layer MLP; all hidden layers are 100 neurons wide, with a skip connection to the middle layer.

Notice that both $h_{\mathcal{X}}(\boldsymbol{x})$ and its signed version are local minima of the loss in equation 2. These local minima are stable in the sense that there is an energy barrier when moving from one to the other. For example, to get to a solution as in Figure 2(b) from the solution in Figure 2(d) one needs to flip the sign in the interior or exterior of the region defined by the black line. Changing the sign continuously will result in a considerable increase to the SAL loss value.

We elaborate on our initialization method, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$, that in practice favors the signed version of $h_{\mathcal{X}}$ in the next section.

4. Geometric network initialization

A key aspect of our method is a proper, geometrically motivated initialization of the network's parameters. For MLPs, equations 8-9, we develop an initialization of its parameters, $\boldsymbol{\theta} = \boldsymbol{\theta}^0$, so that $f(\boldsymbol{x}; \boldsymbol{\theta}^0) \approx \varphi(\|\boldsymbol{x}\| - r)$, where $\|\boldsymbol{x}\| - r$ is the signed distance function to an r-radius sphere. The following theorem specify how to pick $\boldsymbol{\theta}^0$ to achieve this:

Theorem 1. *Let f be an MLP (see equations 8-9). Set, for $1 \leq i \leq \ell$, $\boldsymbol{b}_i = 0$ and $\boldsymbol{W}_i$ i.i.d. from a normal distribution $\mathcal{N}(0, \frac{\sqrt{2}}{\sqrt{d_i^{out}}})$; further set $\boldsymbol{w} = \frac{\sqrt{\pi}}{\sqrt{d_\ell^{out}}}\mathbf{1}$, $c = -r$. Then, $f(\boldsymbol{x}) \approx \varphi(\|x\| - r)$.*

Figure 3 depicts level-sets (zero level-sets in bold) using the initialization of Theorem 1 with the same 8-layer MLP (using $\varphi(a) = a$) and increasing width of 100, 200, and 2000 neurons in the hidden layers. Note how the approximation $f(\boldsymbol{x}; \boldsymbol{\theta}^0) \approx \|\boldsymbol{x}\| - r$ improves as the layers' width increase, while the sphere-like (in this case circle-like) zero level-set remains topologically correct at all approximation levels.

The proof to Theorem 1 is provided in the supplementary material; it is a corollary of the following theorem, showing how to chose the initial weights for a single hidden layer network:

Theorem 2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be an MLP with ReLU activation, ν, and a single hidden layer. That is, $f(\boldsymbol{x}) = \boldsymbol{w}^T \nu(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) + c$, where $\boldsymbol{W} \in \mathbb{R}^{d^{out} \times d}$, $\boldsymbol{b} \in \mathbb{R}^{d^{out}}$, $\boldsymbol{w} \in \mathbb{R}^{d^{out}}$, $c \in \mathbb{R}$ are the learnable parameters. If $\boldsymbol{b} = 0$, $\boldsymbol{w} = \frac{\sqrt{2\pi}}{\sigma d^{out}}\mathbf{1}$, $c = -r$, $r > 0$, and all entries of $\boldsymbol{W}$ are i.i.d. normal $\mathcal{N}(0, \sigma^2)$ then $f(\boldsymbol{x}) \approx \|\boldsymbol{x}\| - r$. That is, f is approximately the signed distance function to a $d-1$ sphere of radius r in $\mathbb{R}^d$, centered at the origin.*

5. Properties

5.1. Plane reproduction

Plane reproduction is a key property to surface approximation methods since, in essence, surfaces are locally planar, *i.e.*, have an approximating tangent plane almost everywhere [16]. In this section we provide a theoretical justification to SAL by proving a plane reproduction property. We first show this property for a linear model (*i.e.*, a single layer MLP) and then show how this implies local plane reproduction for general MLPs.

The setup is as follows: Assume the input data $\mathcal{X} \subset \mathbb{R}^d$ lies on a hyperplane $\mathcal{X} \subset \mathcal{P}$, where $\mathcal{P} = \left\{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{n}^T\boldsymbol{x} + c = 0\right\}$, $\boldsymbol{n} \in \mathbb{R}^d$, $\|\boldsymbol{n}\| = 1$, is the normal to the plane, and consider a linear model $f(\boldsymbol{x}; \boldsymbol{w}, b) = \varphi(\boldsymbol{w}^T\boldsymbol{x} + b)$. Furthermore, we make the assumption that the distribution $D_{\mathcal{X}}$ and the distance $h_{\mathcal{X}}$ are invariant to rigid transformations, which is common and holds in all cases considered in this paper. We prove existence of critical weights $(\boldsymbol{w}^*, b^*)$ of the loss in equation 2, and for which the zero level-set of f, $f(\boldsymbol{x}; \boldsymbol{w}^*, b^*) = 0$, reproduces $\mathcal{P}$:

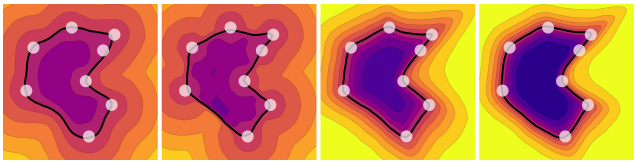


Figure 4: Advanced epochs of the neural level-sets from Figure 2. The limit in the L^0 case (two right images) is an inside/outside indicator function, while for the L^2 case (two left images) it is a signed version of the unsigned L^2 distance.

Theorem 3. *Consider a linear model $f(\boldsymbol{x}; \boldsymbol{\theta}) = \varphi(\boldsymbol{w}^T\boldsymbol{x} + b)$, $\boldsymbol{\theta} = (\boldsymbol{w}, b)$, with a strong non-linearity $\varphi : \mathbb{R} \to \mathbb{R}$. Assume the data $\mathcal{X}$ lies on a plane $\mathcal{P} = \left\{\boldsymbol{x} | \boldsymbol{n}^T\boldsymbol{x} + c = 0\right\}$, i.e., $\mathcal{X} \subset \mathcal{P}$. Then, there exists $\alpha \in \mathbb{R}_+$ so that $(\boldsymbol{w}^*, b^*) = (\alpha\boldsymbol{n}, \alpha c)$ is a critical point of the loss in equation 2.*

This theorem can be applied locally when optimizing a general MLP (equation 8) with SAL to prove local plane reproduction. See supplementary for more details.

5.2. Convergence to the limit signed function

The SAL loss pushes the neural implicit function f towards a signed version of the unsigned distance function $h_{\mathcal{X}}$. In the L^0 case it is the inside/outside indicator function of the surface, while for L^2 it is a signed version of the Euclidean distance to the data $\mathcal{X}$. Figure 4 shows advanced epochs of the 2D experiment in Figure 2; note that the f in these advanced epochs is indeed closer to the signed version of the respective $h_{\mathcal{X}}$. Since the indicator function and the *signed* Euclidean distance are discontinuous across the surface, they potentially impose quantization errors when using standard contouring algorithms, such as Marching Cubes [26], to extract their zero level-set. In practice, this phenomenon is avoided with a standard choice of stopping criteria (learning rate and number of iterations). Another potential solution is to add a regularization term to the SAL loss; we mark this as future work.

6. Experiments

6.1. Surface reconstruction

The most basic experiment for SAL is reconstructing a surface from a single input raw point cloud (without using any normal information). Figure 5 shows surface reconstructions based on four raw point clouds provided in [21] with three methods: ball-pivoting [6], variation-implicit reconstruction [21], and SAL based on the L^0 distance, *i.e.*,

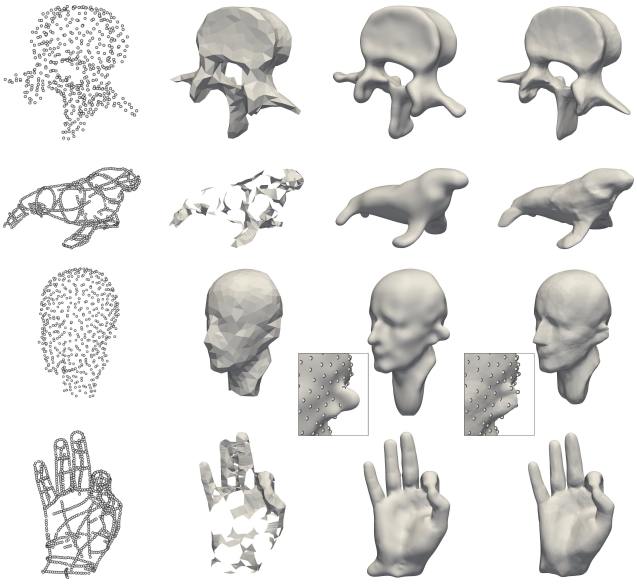


Figure 5: Surface reconstruction from (un-oriented) point cloud. From left to right: input point cloud; ball-pivoting reconstruction [6]; variational-implicit reconstruction [21]; SAL reconstruction (ours).

optimizing the loss described in equation 7 with $\ell = 1$. The only parameter in this loss is σ which we set for every point in $\boldsymbol{x} \in \mathcal{X}$ to be the distance to the 50-th nearest point in the point cloud $\mathcal{X}$. We used an 8-layer MLP, $f : \mathbb{R}^3 \times \mathbb{R}^m \to \mathbb{R}$, with 512 wide hidden layers and a single skip connection to the middle layer (see supplementary material for more implementation details). As can be visually inspected from the figure, SAL provides high fidelity surfaces, approximating the input point cloud even for challenging cases of sparse and irregular input point clouds.

6.2. Learning shape space from raw scans

In the main experiment of this paper we trained on the D-Faust scan dataset [7], consisting of approximately 41k raw scans of 10 humans in multiple poses[2]. Each scan is a triangle soup, $\mathcal{X}_i$, where common defects include holes, ghost geometry, and noise, see Figure 1 for examples.

Architecture. To learn the shape representations we used a modified variational encoder-decoder [24], where the encoder $(\boldsymbol{\mu}, \boldsymbol{\eta}) = g(\boldsymbol{X}; \boldsymbol{\theta}_1)$ is taken to be PointNet [31] (specific architecture detailed in supplementary material), $\boldsymbol{X} \in \mathbb{R}^{n \times 3}$ is an input point cloud (we used $n = 128^2$), $\boldsymbol{\mu} \in \mathbb{R}^{256}$ is the latent vector, and $\boldsymbol{\eta} \in \mathbb{R}^{256}$ represents a diagonal covariance matrix by $\boldsymbol{\Sigma} = \operatorname{diag} \exp \boldsymbol{\eta}$. That is, the encoder takes in a point cloud $\boldsymbol{X}$ and outputs a probability measure $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The point cloud is drawn uniformly at random from the scans, *i.e.*, $\boldsymbol{X} \sim \mathcal{X}_i$. The decoder is the implicit representation $f(\boldsymbol{x}; \boldsymbol{w}, \boldsymbol{\theta}_2)$ with the addition of a latent vector $\boldsymbol{w} \in \mathbb{R}^{256}$. The architecture of f is taken to be the 8-layer MLP, as in Subsection 6.1.

[2]Due to the dense temporal sampling in this dataset we experimented with a 1:5 sample.

Loss. We use SAL loss with L^2 distance, *i.e.*, $h_2(\boldsymbol{z}) = \min_{\boldsymbol{x} \in \mathcal{X}_i} \|\boldsymbol{z} - \boldsymbol{x}\|_2$ the unsigned distance to the triangle soup $\mathcal{X}_i$, and combine it with a variational auto-encoder type loss [24]:

$$\operatorname{Loss}(\boldsymbol{\theta}) = \sum_i \mathbb{E}_{\boldsymbol{X} \sim \mathcal{X}_i} \left[\operatorname{loss}_{\mathrm{R}}(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\mu}\|_1 + \|\boldsymbol{\eta} + \mathbf{1}\|_1 \right]$$

$$\operatorname{loss}_{\mathrm{R}}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{z} \sim \mathcal{N}_\sigma(\mathcal{X}_i), \boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \big| |f(\boldsymbol{z}; \boldsymbol{w}, \boldsymbol{\theta}_2)| - h_2(\boldsymbol{z}) \big|,$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, $\|\cdot\|_1$ is the 1-norm, $\|\boldsymbol{\mu}\|_1$ encourages the latent prediction $\boldsymbol{\mu}$ to be close to the origin, while $\|\boldsymbol{\eta} + \mathbf{1}\|_1$ encourages the variances $\boldsymbol{\Sigma}$ to be constant $\exp(-1)$; together, these enforce a regularization on the latent space. λ is a balancing weight chosen to be 10^{-3}.

Baseline. We compared versus three baseline methods. First, AtlasNet [19], one of the only existing algorithms for learning a shape collection from raw point clouds. AtlasNet uses a parametric representation of surfaces, which is straight-forward to sample. On the down side, it uses a collection of patches that tend to not overlap perfectly, and their loss requires computation of closest points between the generated and input point clouds which poses a challenge for learning large point clouds. Second, we approximate a signed distance function, $\bar{h}_2$, to the data $\mathcal{X}_i$ in two different ways, and regress them using an MLP as in DeepSDF [30]; we call these methods SignReg. Note that Occupancy Networks [28] and [12] regress a different signed distance function and perform similarly.

To approximate the signed distance function, $\bar{h}_2$, we first tried using a state of the art surface reconstruction algorithm [23] to produce watertight manifold surfaces. However, only 28684 shapes were successfully reconstructed (69% of the dataset), making this option infeasible to compute $\bar{h}_2$. We have opted to approximate the signed distance function similar to [20] with $\bar{h}_2(\boldsymbol{z}) = \boldsymbol{n}_*^T(\boldsymbol{z} - \boldsymbol{x}_*)$, where $\boldsymbol{x}_* = \arg\min_{\boldsymbol{x} \in \mathcal{X}_i} \|\boldsymbol{z} - \boldsymbol{x}\|_2$ is the closest point to $\boldsymbol{z}$ in $\mathcal{X}_i$ and $\boldsymbol{n}_*$ is the normal at $\boldsymbol{x}_* \in \mathcal{X}_i$. To approximate the normal $\boldsymbol{n}_*$ we tested two options: (i) taking $\boldsymbol{n}^*$ directly from the original scan $\mathcal{X}_i$ with its original orientation; and (ii) using local normal estimation using Jets [10] followed by consistent orientation procedure based on minimal spanning tree using the CGAL library [1].

Table 1 and Figure 6 show the result on a random 75%-25% train-test split on the D-Faust raw scans. We report the 5%, 50% (median), and 95% percentiles of the Chamfer distances between the surface reconstructions and the

Figure 6: Reconstruction of the test set from D-Faust scans. Left to right in each column: input test scan, SAL (our) reconstruction, AtlasNet [19] reconstruction, and SignReg - signed regression with approximate Jet normals.

		Registrations			Scans		
	Method	5%	Median	95%	5%	Median	95%
Train	AtlasNet[19]	0.09	0.15	0.27	**0.05**	0.09	0.18
	Scan normals	2.53	43.99	292.59	2.63	44.86	257.37
	Jet normals	1.72	30.46	513.34	1.65	31.11	453.43
	SAL (ours)	**0.05**	**0.09**	**0.2**	**0.05**	**0.06**	**0.09**
Test	AtlasNet[19]	0.1	0.17	0.37	**0.05**	0.1	0.22
	Scan normals	3.45	45.03	294.15	3.21	277.36	45.03
	Jet normals	1.88	31.05	489.35	1.76	30.89	462.85
	SAL (ours)	**0.07**	**0.12**	**0.35**	**0.05**	**0.08**	**0.16**

Table 1: Reconstruction of the test set from D-Faust scans. We log the Chamfer distances of the reconstructed surfaces to the raw scans (one-sided), and ground-truth registrations; we report the 5-th, 50-th, and 95-th percentile. Numbers are reported $*10^3$.

raw scans (one-sided Chamfer from reconstruction to scan), and ground truth registrations. The SAL and SignReg reconstructions were generated by a forward pass $(\boldsymbol{\mu}, \boldsymbol{\eta}) = g(\boldsymbol{X}; \boldsymbol{\theta}_1)$ of a point cloud $\boldsymbol{X} \subset \mathcal{X}_i$ sampled from the raw unseen scans, yielding an implicit function $f(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\theta}_2)$. We used the Marching Cubes algorithm [26] to mesh the zero level-set of this implicit function. Then, we sampled uniformly 30K points from it and compute the Chamfer Distance.

Generalization to unseen data. In this experiment we test our method on two different scenarios: (i) generating shapes of unseen humans; and (ii) generating shapes of unseen poses. For the unseen humans experiment we trained on 8 humans (4 females and 4 males), leaving out 2 humans for test (one female and one male). For the unseen poses experiment, we randomly chose two poses of each human as a test set. To further improve test-time shape representations, we also further optimized the latent $\boldsymbol{\mu}$ to better approximate the input test scan $\mathcal{X}_i$. That is, for each test scan $\mathcal{X}_i$, after the forward pass $(\boldsymbol{\mu}, \boldsymbol{\eta}) = g(\boldsymbol{X}; \boldsymbol{\theta}_2)$ with $\boldsymbol{X} \subset \mathcal{X}_i$, we further optimized loss_R as a function of $\boldsymbol{\mu}$ for 800 further iterations. We refer to this method as latent optimization.

Table 2 demonstrates that the latent optimization method further improves predictions quality, compared to a single forward pass. In 7 and 8, we demonstrate few representatives examples, where we plot left to right in each column: input test scan, SAL reconstruction with forward pass alone, and SAL reconstruction with latent optimization. Failure cases are shown in the bottom-right. Despite the little variability of humans in the training dataset (only 8 humans), 7 shows that SAL can usually fit a pretty good human shape to the unseen human scan using a single forward pass reconstruction; using latent optimization further improves the approximation as can be inspected in the different examples in this figure.

Figure 8 shows how a single forward reconstruction is able to predict the pose correctly, where latent optimization improves the prediction in terms of shape and pose.

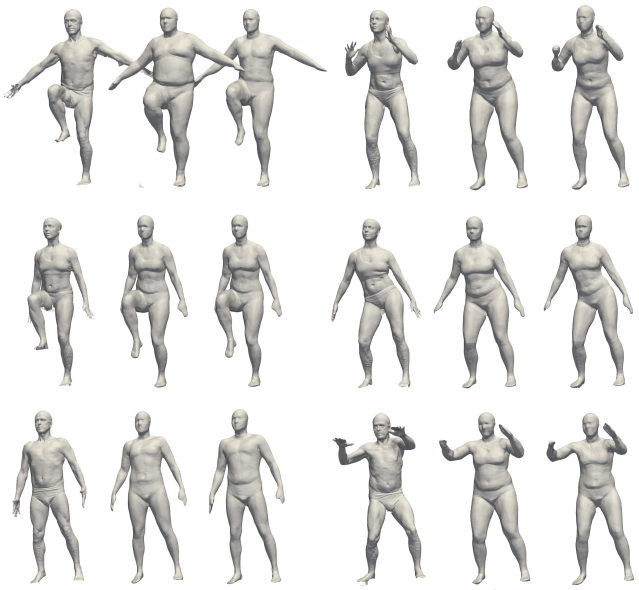


Figure 7: Reconstruction of unseen humans scans. Each column from left to right: unseen human scan, SAL reconstruction with a single forward pass, SAL reconstruction with latent optimization. Bottom-right shows failure.

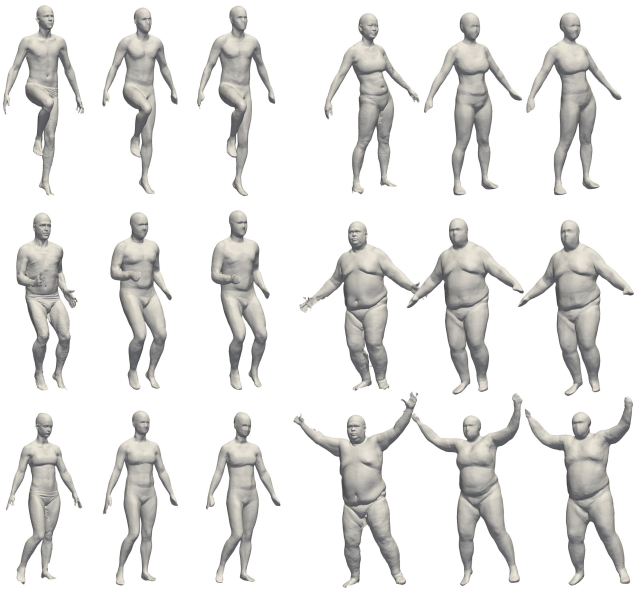


Figure 8: Reconstruction of unseen pose scans. Each column from left to right: unseen pose scan, SAL reconstruction with a single forward pass, SAL reconstruction with latent optimization. Bottom-right shows failure.

Limitations. SAL's limitation is mainly in capturing thin structures. Figure 9 shows reconstructions (obtained similarly to 6.1) of a chair and a plane from the ShapeNet [11] dataset; note that some parts in the chair back and the plane wheel structure are missing.

		Registrations			Scans		
	Method	5%	Median	95%	5%	Median	95%
Train	SAL (Pose)	0.08	0.12	0.25	0.05	0.07	0.1
	SAL (Human)	0.06	0.09	0.18	0.04	0.06	0.09
Test	SAL (Pose)	0.11	0.37	2.26	0.07	0.18	0.93
	SAL + latent opt. (Pose)	0.08	0.16	1.12	0.05	0.09	0.19
	SAL (Human)	0.26	0.75	4.99	0.14	0.34	1.53
	SAL + latent opt. (Human)	0.12	0.3	3.05	0.07	0.14	0.49

Table 2: Reconstruction of the unseen human and pose from D-Faust scans. We log the Chamfer distances of the reconstructed surfaces to the raw scans (one-sided), and ground-truth registrations; we report the 5-th, 50-th, and 95-th percentile. Numbers are reported $*10^3$.

Figure 9: Failure in capturing thin structures. In each pair: ground truth model (left), and SAL reconstruction (right).

7. Conclusions

We introduced SAL: Sign Agnostic Learning, a deep learning approach for processing raw data without any preprocess or need for ground truth normal data or inside/outside labeling. We have developed a geometric initialization formula for MLPs to approximate the signed distance function to a sphere, and a theoretical justification proving planar reproduction for SAL. Lastly, we demonstrated the ability of SAL to reconstruct high fidelity surfaces from raw point clouds, and that SAL easily integrates into standard generative models to learn shape spaces from raw geometric data. One limitation of SAL was mentioned in Section 5, namely the stopping criteria for the optimization.

Using SAL in other generative models such as generative adversarial networks could be an interesting follow-up. Another future direction is global reconstruction from partial data. Combining SAL with image data also has potentially interesting applications. We think SAL has many exciting future work directions, progressing geometric deep learning to work with unorganized, raw data.

Acknowledgments

The research was supported by the European Research Council (ERC Consolidator Grant, "LiftMatch" 771136), the Israel Science Foundation (Grant No. 1830/17) and by a research grant from the Carolito Stiftung (WAIC).

References

[1] Pierre Alliez, Simon Giraudot, Clément Jamin, Florent Lafarge, Quentin Mérigot, Jocelyn Meyron, Laurent Saboret, Nader Salman, and Shihao Wu. Point set processing. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.0 edition, 2019. 6

[2] Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. Controlling neural level sets. *arXiv preprint arXiv:1905.11911*, 2019. 1, 2

[3] Timur Bagautdinov, Chenglei Wu, Jason Saragih, Pascal Fua, and Yaser Sheikh. Modeling facial geometry using compositional vaes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2018. 2

[4] Heli Ben-Hamu, Haggai Maron, Itay Kezurer, Gal Avineri, and Yaron Lipman. Multi-chart generative surface modeling. In *SIGGRAPH Asia 2018 Technical Papers*, page 215. ACM, 2018. 1, 2

[5] Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*, volume 36, pages 301–329. Wiley Online Library, 2017. 1

[6] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999. 5, 6

[7] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 6

[8] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. *arXiv preprint arXiv:1707.05776*, 2017. 2

[9] Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM, 2001. 2

[10] Frédéric Cazals and Marc Pouget. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, 22(2):121–146, 2005. 6

[11] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 8

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 1, 2, 4, 6

[13] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5868–5877, 2017. 2

[14] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnets: Learnable convex decomposition. *arXiv preprint arXiv:1909.05736*, 2019. 1

[15] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. *arXiv preprint arXiv:1908.04725*, 2019. 1

[16] Manfredo P Do Carmo. *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications, 2016. 3, 5

[17] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. *arXiv preprint arXiv:1904.06447*, 2019. 1, 2

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[19] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 1, 2, 6, 7

[20] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. *Surface reconstruction from unorganized points*, volume 26. ACM, 1992. 6

[21] Zhiyang Huang, Nathan Carr, and Tao Ju. Variational implicit point set surfaces. *ACM Trans. Graph.*, 38(4), July 2019. 3, 5, 6

[22] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006. 2

[23] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29, 2013. 2, 6

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2, 6

[25] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1886–1895, 2018. 2

[26] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM siggraph computer graphics*, volume 21, pages 163–169. ACM, 1987. 5, 7

[27] Haggai Maron, Meirav Galun, Noam Aigerman, Miri Trope, Nadav Dym, Ersin Yumer, Vladimir G Kim, and Yaron Lipman. Convolutional neural networks on surfaces via seamless toric covers. *ACM Trans. Graph.*, 36(4):71–1, 2017. 2

[28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1, 2, 6

[29] Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. Signing the unsigned: Robust surface reconstruction from raw pointsets. In *Computer Graphics Forum*, volume 29, pages 1733–1741. Wiley Online Library, 2010. 2

[30] 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. 1, 2, 4, 6

[31] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 6

[32] Ayan Sinha, Jing Bai, and Karthik Ramani. Deep learning 3d shape surfaces using geometry images. In *European Conference on Computer Vision*, pages 223–240. Springer, 2016. 2

[33] Ayan Sinha, Asim Unmesh, Qixing Huang, and Karthik Ramani. Surfnet: Generating 3d shape surfaces using deep residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6040–6049, 2017. 2

[34] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1955–1964, 2018. 2

[35] Kenshi Takayama, Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Consistently orienting facets in polygon meshes by minimizing the dirichlet energy of generalized winding numbers. *arXiv preprint arXiv:1406.5431*, 2014. 3

[36] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017. 2

[37] Christian Walder, Olivier Chapelle, and Bernhard Schölkopf. Implicit surface modelling as an eigenvalue problem. In *Proceedings of the 22nd international conference on Machine learning*, pages 936–939. ACM, 2005. 2

[38] Christian Walder, Olivier Chapelle, and Bernhard Schölkopf. Implicit surfaces with globally regularised and compactly supported basis functions. In *Advances in Neural Information Processing Systems*, pages 273–280, 2007. 2

[39] Holger Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004. 3

[40] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10130–10139, 2019. 1, 2

[41] 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*, pages 82–90, 2016. 2

[42] Hongyi Xu and Jernej Barbič. Signed distance fields for polygon soup meshes. In *Proceedings of Graphics Interface 2014*, pages 35–41. Canadian Information Processing Society, 2014. 3

[43] Hong-Kai Zhao, Stanley Osher, and Ronald Fedkiw. Fast surface reconstruction using the level set method. In *Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 194–201. IEEE, 2001. 2