SAL++: Sign Agnostic Learning with Derivatives

Matan Atzmon and Yaron Lipman
Weizmann Institute of Science

Abstract. Learning 3D geometry directly from raw data, such as point clouds, triangle soups, or un-oriented meshes is still a challenging task that feeds many downstream computer vision and graphics applications. In this paper we introduce SAL++: a method for learning implicit neural representations of shapes directly from such raw data. We build upon the recent sign agnostic learning (SAL) approach and generalize it to include derivative data in a sign agnostic manner. In more detail, given the unsigned distance function to the input raw data, we suggest a novel sign agnostic regression loss, incorporating both pointwise values and gradients of the unsigned distance function. Optimizing this loss leads to a signed implicit function solution, the zero level set of which is a high quality, valid manifold approximation to the input 3D data. We demonstrate the efficacy of SAL++ by shape space learning from two challenging datasets: ShapeNet [9] that contains inconsistent orientation and non-manifold meshes, and D-Faust [8] that contains raw 3D scans (triangle soups). On both these datasets we present state of the art results.

Keywords: implicit neural representations, signed distance function, learning 3D shapes, sign agnostic learning

1 Introduction

Recently, neural networks (NN) have been used for representing and reconstructing 3D surfaces. Current NN-based 3D learning approaches differ in two aspects: the choice of surface representation, and the supervision method. Common representations of surfaces include using NN as parameteric charts of surfaces [19, 39]; volumetric implicit function representation defined over regular grids [40, 36, 21]; and NN used directly as volumetric implicit functions [31, 29, 11], referred henceforth as implicit neural representations. Supervision methods include regression of known or approximated volumetric implicit representations [31, 29, 11], regression directly with raw 3D data [5, 17, 5], and differentiable rendering using 2D data (i.e., images) supervision [30, 26, 34].

The goal of this paper is to introduce SAL++, a method for learning implicit neural representations of surfaces directly from raw 3D data. The benefit in learning directly from raw data, e.g., non-oriented models or triangle soups (e.g., [9]) and raw scans (e.g., [8]), is avoiding the need for a ground truth signed
distance representation of all train surfaces for supervision. This allows working with complex models with inconsistent normals and/or missing parts. In Figure 1 we show reconstructions of zero level sets of SAL++ learned implicit neural representations of car models from the ShapeNet dataset [9] with variational auto-encoder; notice the high detail level and the interior, which would not have been possible with, e.g., previous data pre-processing techniques using renderings of visible parts.

Our approach improves upon the recent Sign Agnostic Learning (SAL) method [5] and shows that incorporating derivative data in a sign agnostic manner provides a significant improvement in surface approximation and detail. SAL is based on the observation that given an unsigned distance function $h$ to some raw 3D data $X \subset \mathbb{R}^3$, such as a point cloud or a triangle soup, a sign agnostic regression to $h$ will introduce new local minima that are signed versions of $h$; in turn, these signed distance functions can be used as implicit representations of the underlying surface. In this paper we show how the sign agnostic regression loss can be extended to compare both function values $h$ and derivatives $\nabla h$ up to a sign. Although the addition of derivative loss does not provide extra (i.e., previously unknown) information on the function $h$ it nevertheless leads to significantly better signed distance functions that capture more detail than the original SAL.

Analyzing theoretical aspects of SAL and SAL++, we observe that both possess the favorable minimal surface property, that is, in areas of missing parts and holes they will prefer zero level sets with minimal area. We justify this property by proving that, in 2D, when restricted to the zero level-set (a curve in this case), the SAL and SAL++ losses would encourage a straight line solution connecting neighboring data points.

We have tested SAL++ on the human raw scan dataset, D-Faust [8], and man-made models, ShapeNet [9], and compared to state of the art methods. In all cases we have used the raw input data $X$ as is and considered the unsigned distance function to $X$, i.e., $h_X$, in the SAL++ loss to produce an approximate signed...
distance function in the form of a neural network. When comparing to ground truth reconstructions we report state of the art results on the D-Faust data-set, striking a balance between approximating details of the scans and avoiding over fitting noise and ghost geometry. For ShapeNet, we are not aware of previous works that reported results when trained directly on the raw data; comparing to state of the art method, trained with pre-processed data we find that our method achieves comparable to superior results.

Summarizing the contributions of this paper:

- Introducing sign agnostic learning for derivative (Hermite) data.
- Identifying and providing some theoretical proof for the minimal surface property of sign agnostic learning.
- Training directly on raw data including unoriented or not consistently oriented triangle soups and raw 3D scans.

2 Previous work

Learning 3D shapes with neural networks and 3D supervision has shown great progress recently. We review related works, where we categorize the existing methods based on their choice of 3D surface representation.

2.1 Parametric representations

The most fundamental surface representation is an atlas, that is a collection of parametric charts \( f : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \) with certain coverage and transition properties \[14\]. \[19\] adapted this idea using neural network to represent a surface as union of such charts; \[39\] improved this construction by introducing better transitions between charts; \[35\] use geometry images \[20\] to represent an entire shape using a single chart; \[28\] use global conformal parameterization for learning surface data; \[7\] use a collection of overlapping global conformal charts for human-shape generative model. The benefit in parametric representations is their ability to easily produce samples of the surface and work directly with raw data (e.g., Chamfer loss); their main drawback is in producing charts that are collectively consistent, of low distortion, and covering the shape.

2.2 Implicit representations

Another approach for representing surfaces is as zero level sets of a function, called an implicit function. There are two popular methods to model implicit volumetric functions with neural networks: i) Convolutional neural network predicting scalar values over a predefined fixed volumetric structure (e.g., grid or octree) in space \[36,40\]; and ii) Multilayer Perceptron of the form \( f : \mathbb{R}^3 \rightarrow \mathbb{R} \) defining a continuous volumetric function \[31,29,11\].
Supervision. Currently, neural networks are trained to be implicit function representations with two types of supervision. (i) regression of samples taken from a known or pre-computed implicit function representation such as occupancy function \[29,11\] or a signed distance function \[31\]. (ii) working with raw 3D supervision, by particle methods relating points on the level sets to the model parameters \[4\], or using sign agnostic losses \[5\].

2.3 Primitives

Another type of representation is to learn shapes as composition or unions of a family of primitives. \[24\] represent a shape using a parametric collection of primitives. \[16,15\] use a collection of axis-aligned Gaussians and learn consistent shape decompositions. \[10\] suggests a differentiable Binary Space Partitioning tree (BSP-tree) for representing shapes. \[13\] combines points and charts representations to learn basic shape structures. \[12\] represents a shape as a union of convex sets. \[38\] learn cites of Voronoi cells for implicit shape representation.

2.4 Template fitting

Lastly, several methods learn 3D shapes of a certain class (e.g., humans) by learning the deformation from a template model. Classical methods use matching techniques and geometric loss minimization for non-rigid template matching \[123\]. \[18\] use an auto-encoder architecture and Chamfer distance to match target shapes. \[25\] use graph convolutional autoencoder to learn deformable template for shape completion.

3 Method

Given raw geometric input data \(\mathcal{X} \subset \mathbb{R}^3\), e.g., a triangle soup, our goal is to find a multilayer perceptron (MLP) \(f : \mathbb{R}^3 \times \mathbb{R}^m \rightarrow \mathbb{R}\) whose zero level-set,

\[
\mathcal{S} = \{ x \in \mathbb{R}^3 \mid f(x; \theta) = 0 \}
\]

is a manifold surface that approximates \(\mathcal{X}\).

Sign agnostic learning. Similarly to SAL, our approach is to consider the (readily available) unsigned distance function to the raw input geometry,

\[
h(y) = \min_{x \in \mathcal{X}} \|y - x\|
\]

and perform sign agnostic regression to get a signed version \(f\) of \(h\). SAL uses a loss of the form

\[
\text{loss}(\theta) = \mathbb{E}_{x \sim D} \tau(f(x; \theta), h(x)),
\]

where \(D\) is some probability distribution, and \(\tau\) is an unsigned similarity. That is, \(\tau(a, b)\) is measuring the difference between scalars \(a, b \in \mathbb{R}\) up-to a sign. For example

\[
\tau(a, b) = ||a| - b|
\]
is an example that is used in [5]. The key property of the sign agnostic loss in equation 3 is that, with proper weights initialization $\theta_0$, it introduces new local minima $f$ of signed distance function which in absolute value are similar to $h$. In turn, the zero level set $S$ of $f$ is a valid manifold describing the data $X$.

**Sign agnostic learning with derivatives.** Our goal is to generalize the SAL loss (equation 3) to include derivative data of $h$ and show that optimizing this loss provides implicit neural representations, $S$, that enjoy better accuracy with respect to the underlying geometry $X$.

The first step in generalizing equation 3 is designing an unsigned similarity measure $\tau$ for vector valued functions. We suggest two such measures: First, notice that equation 4 can be written as

$$\tau(a, b) = \min \{ |a - b|, |a + b| \}, a, b \in \mathbb{R},$$

and therefore generalizes to vectors $a, b \in \mathbb{R}^d$,

$$\tau(a, b) = \min \{ \| a - b \|, \| a + b \| \},$$

where $\| \cdot \|$ is some norm; we use the $L_2$ norm, $\|a\| = \|a\|_2 = \sqrt{a^T a}$. A second option is

$$\tau(a, b) = |\sin \alpha|$$

where $\alpha = \angle(a, b)$ the angle between $a, b$. Note that $\tau(-a, b) = |\sin (\pi - \alpha)| = |\sin \alpha| = \tau(a, b)$ so $\tau$ is unsigned. The difference between the two options of $\tau$ is that equation 6 only penalizes the difference in directions of $a, b$, while equation 5 penalizes both angle and length differences.

We define the SAL++ loss:

$$\text{loss}(\theta) = \mathbb{E}_{x \sim D} \tau(f(x; \theta), h(x)) + \lambda \mathbb{E}_{x \sim D'} \tau(\nabla_x f(x; \theta), \nabla_x h(x))$$

where $\lambda > 0$ is a parameter, $D'$ is a probability distribution, and $\nabla_x f(x; \theta), \nabla_x h(x)$ are the gradients $f, h$ (resp.) with respect to their input $x$.

In Figure 2 we show the unsigned distance $h$ to an L-shaped curve (left), and the level sets of the MLPs optimized with the SAL++ loss (middle) and the SAL loss (right); note that SAL++ loss reconstructed the sharp features (i.e., corners) of the shape and the level sets of $h$, while SAL loss smoothed them out; the implementation details of this experiment can be found in the supplementary material.

**Minimal surface property.** We show that the SAL and SAL++ losses possess a minimal surface property [42], that is, they strives to minimize surface area of missing parts. For example, Figure 4 shows the unsigned distance to a curve with a missing segment (left), and the zero level sets of MLPs optimized with SAL++ loss (middle), and SAL loss (right). Note that in both cases the zero level set in the missing part area is the minimal length curve (i.e., a line) connecting the end points of that missing part. SAL++ also preserves sharp features of the rest of the shape.
Fig. 2. Sign agnostic learning of an unsigned distance function to an L shape (left). Red colors depict positive values, and blue-green colors depict negative values. In the middle the result of optimizing the SAL++ loss (equation 7); on the right, the result of SAL loss (equation 3). Note that SAL++ better preserves sharp features of the shape and the isolevels.

Fig. 3. Minimal surface property in 2D.

We will provide a theoretical justification to this property in the 2D case: we consider a geometry defined by two points in the plane, \( \mathcal{X} = \{ \mathbf{x}_1, \mathbf{x}_2 \} \subset \mathbb{R}^2 \) and possible solutions where the zero level set curve \( \mathcal{S} \) is connecting \( \mathbf{x}_1 \) and \( \mathbf{x}_2 \). We prove that among a class of curves \( \mathcal{U} \) connecting \( \mathbf{x}_1 \) and \( \mathbf{x}_2 \), the straight line minimizes the losses in equation 3 and equation 7 restricted to the curves, when assuming e.g., uniform distributions \( D, D' \). We assume (without losing generality) that \( \mathbf{x}_1 = (0, 0)^T \), \( \mathbf{x}_2 = (\ell, 0)^T \) and consider curves \( \mathbf{u} \in \mathcal{U} \) defined by \( \mathbf{u}(t) = (s, t(s))^T \), where \( s \in [0, \ell] \), and \( t : \mathbb{R} \to \mathbb{R} \) is some differentiable function such that \( t(0) = 0 = t(\ell) \), see Figure 3. We now prove

**Theorem 1.** Let \( \mathcal{X} = \{ \mathbf{x}_1, \mathbf{x}_2 \} \subset \mathbb{R}^2 \), and the family of curves \( \mathcal{U} \) connecting \( \mathbf{x}_1 \) and \( \mathbf{x}_2 \). Furthermore, let \( \text{loss}_{\text{SAL}}(\mathbf{u}) \) and \( \text{loss}_{\text{SAL}++}(\mathbf{u}) \) denote the losses in equation 3 and equation 7 (resp.) when restricted to \( \mathbf{u} \) with uniform distributions \( D, D' \). Then in both cases the straight line, i.e., the curve \( \mathbf{u}(s) = (s, 0) \), is the strict global minimizer of these losses.

**Proof.** First, consider the SAL loss, equation 3 with a uniform distribution \( D \) in some bounding box, and restrict it to the curve \( \mathbf{u} \) (i.e., where \( f \) vanishes). The unsigned distance function in this case is

\[
h(s, t) = \begin{cases} \sqrt{s^2 + t^2} & s \in [0, \ell/2] \\ \sqrt{(s - \ell)^2 + t^2} & s \in (\ell/2, \ell] \end{cases}
\]

From symmetry it is enough to consider only the first half of the curve, i.e., \( s \in [0, \ell/2] \). The SAL loss takes the form

\[
\text{loss}_{\text{SAL}}(\mathbf{u}) = \int_0^{\ell/2} \sqrt{s^2 + t^2} \sqrt{1 + t^2} \, ds,
\]
Fig. 4. Minimal surface property: using SAL++ (middle) and SAL (right) with the input unsigned distance function of a curve with a missing part (left) leads to a solution (black line, middle and right) with approximately minimal length in the missing part area. Note that the SAL++ solution also preserves sharp features of the original shape, better than SAL.

where \( \sqrt{1 + \dot{t}^2} \, ds \) is the length element on the curve \( u \), and \( \tau(f(s; t; \theta), h(s, t)) = |h(s, t)| = \sqrt{s^2 + t^2} \), since \( f(s; t; \theta) = 0 \) over the curve \( u \). Plugging \( t(s) \equiv 0 \) in \( \text{loss}_{\text{SAL}}(u) \) we see that the curve \( u = (s, 0)^T \), namely the straight line curve from \( x_1 \) to \( 0.5(x_1 + x_2) \) is a strict global minimizer of \( \text{loss}_{\text{SAL}}(u) \). Similar argument on \( s \in [\ell/2, \ell] \) finish the proof for the SAL case.

For the SAL++ case we prove the claim for the sign agnostic similarity measure in equation 6. Note that \( \dot{u} = (1, \dot{t})^T \) and a normal direction to this zero level set is \( \dot{u}^\perp = (-\dot{t}, 1)^T \). Therefore,

\[
\tau(\dot{u}^\perp, \nabla h(s, t)) = |\sin \alpha| = \frac{\left| \det \begin{pmatrix} -\dot{t} & s \\ 1 & t \end{pmatrix} \right|}{\sqrt{1 + \dot{t}^2 \sqrt{s^2 + t^2}}},
\]

where the last equality can be checked by differentiating \( \|s(t)\| \) w.r.t. \( s \). Plugging this in the derivative loss term of SAL++ loss, restricted to \( u \), and considering as above half the segment \( s \in [0, \ell/2] \), we get

\[
\text{loss}_{\text{SAL++}}(u) - \text{loss}_{\text{SAL}}(u) \geq \lambda \int_0^{\ell/2} \frac{d}{ds} \|s(t)\| \, ds = \lambda \left\| \left( \frac{\ell}{2}, t \left( \frac{\ell}{2} \right) \right) \right\|,
\]

where the first inequality is due to the removal of the absolute value around \( \frac{d}{ds} \|s(t)\| \). This bound implies that the curve \( u = (s, 0)^T \) is a minimizer of this loss (although not necessarily a strict one as is the case for the SAL loss). However, combining the result for the SAL loss we get that the straight line curve \( u = (s, 0)^T \) is a strict global minimizer of \( \text{loss}_{\text{SAL++}}(u) = (\text{loss}_{\text{SAL++}}(u) - \text{loss}_{\text{SAL}}(u)) + \text{loss}_{\text{SAL}}(u) \).

\( \square \)
4 Experiments

We tested SAL++ on the task of shape space learning from raw 3D data. We experimented with two different datasets: i) ShapeNet dataset [9], containing synthetic 3D Meshes; and ii) D-Faust dataset [8] containing raw 3D scans.

Evaluation metrics. We use the following metrics to measure similarity between shapes:

\[
d_C(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{2} (d^{\rightarrow}_C(\mathcal{X}_1, \mathcal{X}_2) + d^{\rightarrow}_C(\mathcal{X}_2, \mathcal{X}_1)) \tag{8}
\]

where

\[
d^{\rightarrow}_C(\mathcal{X}_1, \mathcal{X}_2) = \frac{1}{|\mathcal{X}_1|} \sum_{x_1 \in \mathcal{X}_1} \min_{x_2 \in \mathcal{X}_2} \|x_1 - x_2\| \tag{9}
\]

and the sets \(\mathcal{X}_i\) are either point clouds or triangle soups. In addition, to measure similarity of the normals of triangle soups \(\mathcal{T}_1, \mathcal{T}_2\), we define:

\[
d_N(\mathcal{T}_1, \mathcal{T}_2) = \frac{1}{2} (d^{\rightarrow}_N(\mathcal{T}_1, \mathcal{T}_2) + d^{\rightarrow}_N(\mathcal{T}_2, \mathcal{T}_1)), \tag{10}
\]

where

\[
d^{\rightarrow}_N(\mathcal{T}_1, \mathcal{T}_2) = \frac{1}{|\mathcal{T}_1|} \sum_{x_1 \in \mathcal{T}_1} \langle n(x_1), n(\hat{x}_1) \rangle, \tag{11}
\]

where \(\langle a, b \rangle\) is the positive angle between vectors \(a, b \in \mathbb{R}^3\), \(n(x_1)\) denotes the face normal of a point \(x_1\) in triangle soup \(\mathcal{T}_1\), and \(\hat{x}_1\) is the projection of \(x_1\) on \(\mathcal{T}_2\).
### 4.1 ShapeNet

In this experiment we test SAL++ ability to learn a shape space by training on a challenging 3D data such as non-manifold/non-orientable meshes. To this end, we train and evaluate our method on five different categories from the ShapeNet dataset.

**Architecture and loss.** Our method can be easily incorporated into existing shape space learning architectures. We experiment with two architecture: i) Auto-Decoder (AD) suggested in [31]; and the ii) Modified Variational Auto-Encoder (VAE) used in [5]. For both options, the decoder is the implicit representation in equation 1, where $f(x; \theta)$ is taken to be an 8-layer MLP with 512 hidden units in each layer and Softplus activation. In addition, to enable sign agnostic learning we initialize the decoder $f(x; \theta)$ weights using the geometric initialization from [5]. For SAL++ training, we use the loss in equation 7 with $\tau(a, b) = \min\{\|a-b\|, \|a+b\|\}$ and $\lambda = 0.01$. See the supplementary for more details regarding the architecture.

**Results.** Table 1 and Figure 5 show quantitative and qualitative results (resp.) comparing SAL++ and DeepSDF [31] on ShapeNet. As can be read from the table and inspected in the figure, our method, when used with the same auto-decoder as in DeepSDF, compares favorably to DeepSDF’s reconstruction performance on this data. Qualitatively the surfaces produces by SAL++ are smoother, mostly with more accurate sharp features. Figure 1 shows train and test results with the VAE.

**Comparing VAE and AD.** Figure 6 shows a comparison between VAE and AD in reconstruction of a test car model. Note that the AD seems to produce more details than the VAE.

| Category | Sofas Mean | Chairs Mean | Tables Mean | Planes Mean | Lamps Mean |
|----------|-----------|------------|-------------|-------------|------------|
| DeepSDF  | 0.329     | 0.341      | 0.839       | 0.177       | 0.909      |
| SAL++ (VAE) | 0.391   | 0.415      | 0.679       | 0.197       | 1.808      |
| SAL++ (AD) | 0.207   | 0.281      | 0.408       | 0.098       | 0.506      |

*Table 1. ShapeNet quantitative results. We log the mean and median of the Chamfer distances ($d_C$) between the reconstructed 3D surfaces and the ground truth meshes. Numbers are reported $\times 10^3$."

### 4.2 D-Faust

The D-Faust dataset [8] contains raw scans (triangle soups) of 10 humans in multiple poses. There are approximately 41k scans in the dataset. Due to the low variety between adjacent scans, we sample each pose scans at a ratio of 1 : 5. The leftmost column in Figure 7 shows examples of raw scans used for training.
Fig. 6. Comparing reconstruction of a test car model (left) with auto-decoder SAL++ (middle), and variational auto encoder SAL++ (right). Note the auto-decoder is able to produce more details of the test model, e.g. steering wheel and headlights.

For evaluation we use the registrations provided with the data set. Note that the registrations where not used for training.

Architecture and loss. We use the modified variational auto-decoder (VAE) suggested in [5]. For the encoder we take PointNet [33]. The decoder is defined as in the ShapeNet experiment above. Here we used $\lambda = 0.1$.

Results. We evaluate SAL++ on the same train/test split from [5]. We compare SAL++ versus two baselines: SAL [5] and IGR [17]. The work of [17] is concurrent to ours. Table 2 and Figure 7 show quantitative and qualitative results (resp.); although SAL++ does not produces the best test results it is roughly comparable in every measure to the best among the two baselines. This means it produces details comparable to IGR while maintaining the minimal surface property as SAL and not adding undesired surface sheets as IGR; see the figure for visual illustrations of these properties: the high level of details of SAL++ and IGR compared to SAL, and the base added by IGR, avoided by SAL++.

|            | $d_C^*$ (reg., recon.) | $d_C^*$ (reg., recon.) | $d_C^*$ (recon., reg.) | $d_C^*$ (recon., reg.) | $d_N^*$ (scan, recon.) | $d_N^*$ (scan, recon.) |
|------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| Mean       | Median                  | Mean                    | Median                  | Mean                    | Median                  | Mean                    |
| SAL        | 0.261 0.239             | 12.303 12.122           | 0.286 0.188             | 10.41 10.88             | 0.188 0.175             | 9.038 8.825             |
| Train IGR  | 0.256 0.242             | 10.271 10.166           | 3.628 3.502             | 16.894 17.507           | 0.258 0.169             | 5.802 5.622             |
| SAL++      | 0.262 0.242             | 10.455 10.303           | 0.714 0.357             | 10.178 10.475           | 0.184 0.173             | 6.077 5.94              |
| Test IGR   | 0.418 0.328             | 13.21 12.459            | 0.344 0.256             | 11.354 10.522           | 0.429 0.246             | 10.096 9.096            |
| SAL++      | 0.276 0.187             | 10.328 9.822            | 3.806 3.627             | 17.124 17.902           | 0.241 0.11              | 5.829 5.295             |
| Test       | 0.443 0.336             | 11.831 10.932           | 0.671 0.437             | 11.884 10.931           | 0.399 0.257             | 7.973 6.991             |

Table 2. D-Faust quantitative results. We log mean and median of the one-sided Chamfer and normal distances between registration meshes (reg), reconstructions (recon) and raw input scans (scan). Number are reported $\ast 10^2$. 

Fig. 7. D-Faust [8] qualitative results on test examples. Columns from left to right: raw scans (magenta depict back-faces), registrations (not used in training), SAL++, IGR [17], and SAL [5].

4.3 Limitations

Figure 8 shows typical failure cases of our method from the ShapeNet experiment described above. We mainly suffer from two types of failures: First, since inside and outside information is not known (and often not even well defined in ShapeNet models) SAL++ can add surface sheets closing what should be open areas (e.g., the bottom side of the lamp, or holes in the chair). Second, thin structures can be missed (e.g., the electric cord of the lamp on the left).

5 Conclusions

We introduced SAL++, a method for learning implicit neural representations from raw data. The method is based on a generalization of the sign agnostic learning idea to include derivative data. We demonstrated that the addition of a sign agnostic derivative term to the loss improves the approximation power of the resulting signed implicit neural network. In particular, showing improvement in the level of details and sharp features of the reconstructions. Furthermore, we
identify the favorable minimal surface property of the SAL and SAL++ losses and provide a theoretical justification in 2D.

We see two possible venues for future work: First, it is clear that there is room for further improvement in approximation properties of implicit neural representations. Although the results in D-Faust are already close to the input quality, in ShapeNet we still see a gap between input models and their implicit neural representations; this challenge already exists in overfitting a large collection of diverse shapes in the training stage. Improvement can come from adding expressive power to the neural networks, or further improving the training losses; adding derivatives as done in this paper is one step in that direction but does not solve the problem completely. Second, it is interesting to think of applications or settings in which SAL++ can improve the current state of the art. Generative 3D modeling is one concrete option, learning geometry with 2D supervision is another.
References

1. Allen, B., Curless, B., Popović, Z.: Articulated body deformation from range scan data. ACM Transactions on Graphics (TOG) 21(3), 612–619 (2002)
2. Allen, B., Curless, B., Popović, Z.: The space of human body shapes: reconstruction and parameterization from range scans. ACM transactions on graphics (TOG) 22(3), 587–594 (2003)
3. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM SIGGRAPH 2005 Papers, pp. 408–416 (2005)
4. Atzmon, M., Haim, N., Yariv, L., Israelov, O., Maron, H., Lipman, Y.: Controlling neural level sets. arXiv preprint arXiv:1905.11911 (2019)
5. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. arXiv preprint arXiv:1911.10414 (2019)
6. Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M.: Automatic differentiation in machine learning: a survey. The Journal of Machine Learning Research 18(1), 5595–5637 (2017)
7. Ben-Hamu, H., Maron, H., Kezurer, I., Avineri, G., Lipman, Y.: Multi-chart generative surface modeling. ACM Transactions on Graphics (TOG) 37(6), 1–15 (2018)
8. Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic faust: Registering human bodies in motion. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6233–6242 (2017)
9. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
10. Chen, Z., Tagliasacchi, A., Zhang, H.: Bsp-net: Generating compact meshes via binary space partitioning. arXiv preprint arXiv:1911.06971 (2019)
11. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019)
12. Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G., Tagliasacchi, A.: Cvxnets: Learnable convex decomposition. arXiv preprint arXiv:1909.05736 (2019)
13. Deprelle, T., Groueix, T., Fisher, M., Kim, V., Russell, B., Aubry, M.: Learning elementary structures for 3d shape generation and matching. In: Advances in Neural Information Processing Systems. pp. 7433–7443 (2019)
14. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition. Courier Dover Publications (2016)
15. Genova, K., Cole, F., Sud, A., Sarna, A., Funkhouser, T.: Deep structured implicit functions. arXiv preprint arXiv:1912.06126 (2019)
16. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. arXiv preprint arXiv:1904.06447 (2019)
17. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. arXiv preprint arXiv:2002.10099 (2020)
18. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3d-coded: 3d correspondences by deep deformation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 230–246 (2018)
19. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 216–224 (2018)
20. Gu, X., Gortler, S.J., Hoppe, H.: Geometry images. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques. pp. 355–361 (2002)
21. Jiang, Y., Ji, D., Han, Z., Zwicker, M.: Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. arXiv preprint arXiv:1912.07109 (2019)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
24. Li, L., Sung, M., Dubrovina, A., Yi, L., Guibas, L.J.: Supervised fitting of geometric primitives to 3d point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2652–2660 (2019)
25. Litany, O., Bronstein, A., Bronstein, M., Makadia, A.: Deformable shape completion with graph convolutional autoencoders. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1886–1895 (2018)
26. Liu, S., Saito, S., Chen, W., Li, H.: Learning to infer implicit surfaces without 3d supervision. In: Advances in Neural Information Processing Systems. pp. 8293–8304 (2019)
27. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: ACM siggraph computer graphics. vol. 21, pp. 163–169. ACM (1987)
28. Maron, H., Galun, M., Aigerman, N., Trope, M., Dym, N., Yumer, E., Kim, V.G., Lipman, Y.: Convolutional neural networks on surfaces via seamless toric covers. ACM Trans. Graph. 36(4), 71–1 (2017)
29. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)
30. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. arXiv preprint arXiv:1912.07372 (2019)
31. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
32. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
33. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)
34. Saito, S., Huang, Z., Natsume, R., Morishima, S., Kanazawa, A., Li, H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2304–2314 (2019)
35. Sinha, A., Bai, J., Ramani, K.: Deep learning 3d shape surfaces using geometry images. In: European Conference on Computer Vision. pp. 223–240. Springer (2016)
36. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2088–2096 (2017)
37. The CGAL Project: CGAL User and Reference Manual. CGAL Editorial Board, 5.0.2 edn. (2020). https://doc.cgal.org/5.0.2/Manual/packages.html
6 Appendix

6.1 Implementation Details.

Data Preparation. Given some raw 3D data $X$, SAL++ loss (See equation 7) is computed on points and corresponding unsigned distance derivative data $\{h(x)\}_{x \in D}$ and $\{\nabla_x h(x')\}_{x' \in D'}$ sampled from some distributions $D$ and $D'$. In this paper, we chose $D$ by uniformly sampling points $\{y\}$ from $X$ and placing two isotropic Gaussians, $N(y, \sigma^2_1 I)$ and $N(y, \sigma^2_2 I)$ for each $y$. The distribution parameter $\sigma_1$ depends on each point $y$, set to be as the distance of the 50th closest point to $y$, whereas $\sigma_2$ is set to 0.3 fixed. The distribution $D'$ is set to uniform on $X$. Computing the unsigned distance to $X$ is done using the CGAL library [37]. To speed up training, we precomputed for each shape in the dataset, 500K samples of the form $\{h(x)\}_{x \in D}$ and $\{\nabla_x h(x')\}_{x' \in D'}$.

Gradient computation The SAL++ loss requires incorporating the term $\nabla_x f(x; \theta)$ in a differentiable manner. Our computation of $\nabla_x f(x; \theta)$ is based on AUTOMATIC DIFFERENTIATION [6] forward mode. Similarly to [17], $\nabla_x f(x; \theta)$ is constructed as a network consists of layers of the form

$$\nabla_x y^{\ell+1} = \text{diag} \left( \sigma' \left( W_{\ell+1} y^\ell + b_{\ell+1} \right) \right) W_{\ell+1} \nabla_x y^\ell$$

where $y^\ell$ denotes the output of the $\ell$ layer in $f(x; \theta)$ and $\theta = (W_\ell, b_\ell)$ are the learnable parameters.

6.2 Architecture Details

VAE Architecture. Our VAE architecture is based on the one used in [5]. The encoder $g(X; \theta_1)$, where $X \in \mathbb{R}^{N \times 3}$ is the input point cloud, is composed of DeepSets [41] and PointNet [33] layers. Each layer consists of

$$\text{PFC}(d_{\text{in}}, d_{\text{out}}) : X \mapsto \nu(XW + 1b^T)$$

$$\text{PL}(d_{\text{in}}, 2d_{\text{in}}) : Y \mapsto [Y, \max(Y)1]$$
where $[\cdot, \cdot]$ is the concat operation, $W \in \mathbb{R}^{d_{in} \times d_{out}}$ and $b \in \mathbb{R}^{d_{out}}$ are the layer weights and bias and $\nu(\cdot)$ is the pointwise non-linear ReLU activation function. Our encoder architecture is:

\[
\text{PFC}(3, 128) \rightarrow \text{PFC}(128, 128) \rightarrow \text{PL}(128, 256) \rightarrow \text{PFC}(256, 128) \rightarrow \text{PL}(128, 256) \rightarrow \text{PFC}(256, 128) \rightarrow \text{PL}(128, 256) \rightarrow \text{PFC}(256, 256) \rightarrow \text{MaxPool} \times 2 \rightarrow \text{FC}(256, 256),
\]

where $\text{FC}(d_{in}, d_{out}) : x \mapsto \nu(Wx + b)$ denotes a fully connected layer. The final two fully connected layers outputs vectors $\mu \in \mathbb{R}^{256}$ and $\eta \in \mathbb{R}^{256}$ used for parametrization of a multivariate Gaussian $\mathcal{N}(\mu, \text{diag exp } \eta)$ used for sampling a latent vector $z \in \mathbb{R}^{256}$. Our encoder architecture is similar to the one used in [29].

Our decoder $f([x, z]; \theta_2)$ is a composition of 8 layers where the first layer is $\text{FC}(256 + 3, 512)$, middle layers are $\text{FC}(512, 512)$ and the final layer is $\text{Linear}(512, 1)$. Notice that the input for the decoder is $[x, z]$ where $x \in \mathbb{R}^3$ and $z$ is the latent vector. In addition, we add a skip connection between the input to the middle fourth layer. We chose the Softplus with $\beta = 100$ for the non linear activation in the FC layers. For regularization of the latent $z$, we add the following term to training loss

\[
0.001 \times (\|\mu\|_1 + \|\eta + 1\|_1),
\]

similarly to [5].

**Auto-Decoder Architecture.** We use an auto-decoder architecture, similar to the one suggested in [31]. We defined the latent vector $z \in \mathbb{R}^{256}$. The decoder architecture is the same as the one described above for the VAE. For regularization of the latent $z$, we add the following term to the loss

\[
0.001 \times \|z\|_2^2,
\]

similarly to [31].

### 6.3 Training details

We trained our networks using the Adam [22] optimizer, setting the batch size to 64. On each training step the SAL++ loss is evaluated on a random draw of $92^2$ points out of the precomputed 500K samples. For the VAE, we set a fixed learning rate of 0.0005, whereas for the AD we scheduled the learning rate to start from 0.0005 and decrease by a factor of 0.5 every 500 epochs. All models were trained for 3000 epochs. Training was done on 4 Nvidia V-100 GPUs, using PyTorch deep learning framework [32].
For the two dimensional experiments in figures 2 and 4, we have used the same decoder as in the VAE architecture with the only difference that the first layer is FC(2,512) (no concatenation of a latent vector to the 2D input). We optimized using the Adam [22] optimizer, for 5000 epochs. The parameter $\lambda$ in the SAL++ loss was set to 0.1.

### 6.4 Evaluation

Tables 1 and 2 in the main paper report quantitative evaluation of our method, compared to other baselines. The meshing of the learned implicit representation of the form equation 1 was done using the MARCHING CUBES algorithm [27] on a uniform cubical grid of size $[512]^3$. Computing the evaluation metrics $d_C$ and $d_N$ is done on a uniform sample of 30K points from the meshed surface.

### 6.5 More results

We provide additional qualitative results for our method from the experiments in section 4 in the main paper. Figure 9 shows results from our VAE experiment on the D-Faust dataset. The leftmost and rightmost columns are reconstructions obtained by a single forward pass on test (unseen) objects. The middle columns were produced by linear interpolating the leftmost and rightmost latent representations. Notice the ability of SAL++ to generate novel mixed faces, body parts and poses. In addition, figure 10 shows results from our Auto-Decoder experiment on the ShapeNet datasets. Similarly to the above, leftmost and rightmost columns are reconstructions of test objects and middle columns show interpolated latent representation reconstructions.
Fig. 9. Latent interpolation between unseen humans on the D-Faust [8] dataset.
Fig. 10. Latent interpolation between unseen objects from the ShapeNet [9] dataset.