SSR: Semi-supervised Soft Rasterizer for single-view 2D to 3D Reconstruction

Issam Laradji  
McGill University, Element AI  
issam.laradji@gmail.com

Pau Rodríguez  
Element AI  
pau.rodriguez@servicenow.com

David Vazquez  
Element AI  
david.vazquez@servicenow.com

Derek Nowrouzezahrai  
McGill University  
derek@cim.mcgill.ca

Abstract

Recent work has made significant progress in learning object meshes with weak supervision. Soft Rasterization methods have achieved accurate 3D reconstruction from 2D images with viewpoint supervision only. In this work, we further reduce the labeling effort by allowing such 3D reconstruction methods leverage unlabeled images. In order to obtain the viewpoints for these unlabeled images, we propose to use a Siamese network that takes two images as input and outputs whether they correspond to the same viewpoint. During training, we minimize the cross entropy loss to maximize the probability of predicting whether a pair of images belong to the same viewpoint or not. To get the viewpoint of a new image, we compare it against different viewpoints obtained from the training samples and select the viewpoint with the highest matching probability. We finally label the unlabeled images with the most confident predicted viewpoint and train a deep network that has a differentiable rasterization layer. Our experiments show that even labeling only two objects yields significant improvement in IoU for ShapeNet when leveraging unlabeled examples. Code is available at https://github.com/IssamLaradji/SSR.

1. Introduction

Acquiring 3D representations of objects is key in many application domains. For example, these representations can be used in autonomous driving, robotics, remote sensing, and medical treatment [8, 18]. They can also be used to create digital twins of dynamic scenes for deep analysis [13], and to synthesize novel views and scenes as efficient ways to augment datasets [26]. Manually modelling 3D objects requires a significant amount of human effort. Fortunately, we can reduce that effort using models that combine deep learning and differentiable renderers to train models on 2D images of an object and generate a 3D representation of that object [27].

Over the last years, the research community has proposed multiple methods to learn to perform single-view 2D image to 3D reconstruction using either viewpoints, voxels, point-clouds, or silhouettes [3, 11, 12, 38]. However, they assume that the dataset is fully labeled and they do not leverage unlabeled examples which are available in abundance. Such additional data could potentially provide strong signal for the model to boost its 3D reconstruction performance. Although work in this area is still scarce [25, 33, 34], in other areas like classification and semantic segmentation using unlabeled data is a common practice and is proven to work with techniques based on semi- and self-supervision [6, 9, 17, 28, 35, 36, 40].

Semi-supervised methods for single-view 2D image to 3D reconstruction with viewpoint supervision remain largely unexplored. A viewpoint represents the distance, elevation, and azimuth from a set vantage point as shown in Figure 2. The most relevant work is [25] which addresses this task with 3D supervision. Unfortunately, 3D supervision is costly to acquire and it often requires 3D modeling expertise for creating the CAD models of different objects, which could be prohibitive in many cases. Viewpoint supervision on the other hand is easier to acquire. For instance, it is possible to obtain viewpoints from well-calibrated cameras [1, 31]. However, requiring calibration makes it difficult to acquire labeled training data. In other cases, it is possible to train a generic viewpoint inference model [30] that could be used out of the box to get viewpoint estimates from images in the wild. The advantage of these scenarios is that they do not need 3D expertise in the process of learning to construct 3D meshes. Although viewpoints are much easier to acquire than 3D models of an object, their collection is not straightforward for objects in specific domains.

In this paper, we address the problem of 3D reconstruction from a single 2D image without 3D supervision. We
propose SSR, a semi-supervised learning algorithm to leverage unlabeled data and reduce the viewpoint annotation costs such as camera calibration or human expertise. In order to approximate viewpoints from unlabeled data we propose SiamVP, a Siamese network [10] that is trained to take two images with known viewpoint as input and outputs the probability that those images correspond to the same viewpoint. To approximate the viewpoints of unlabeled images, we use SiamVP to compare them against other training images of the same class with known viewpoints. We finally assign the most confident predicted viewpoints to the unlabeled images and train the soft rasterizer on them. In addition, we also include augmentation to our training of SiamVP. Two images of matching viewpoints will still match when they undergo the same affine transformation such as rotation, scale and translation. We empirically show that SSR achieves significant improvement in IoU for ShapeNet [4] when leveraging unlabeled samples even when only two labeled objects are available. In an ablation study, we empirically show the effectiveness of the proposed data augmentation based on rotations for SiamVP.

The contributions of this work can be summarized as follows:

- We establish a novel framework that leverages unlabeled samples for learning to construct 3D meshes from 2D images with viewpoint supervision.
- We show that using augmentations can highly stabilize the training of SiamVP and achieve more accurate viewpoint predictions.
- We show that SSR provides a consistent boost of performance over SoftRas for different amounts of labeled data with respect to IoU on ShapeNet.

2. Related Work

This work intersects with the topics of Reconstruction with 3D supervision, Reconstruction with Viewpoint Supervision, and Semi- and Self-Supervised Learning which we will discuss in detail in the following paragraphs.

Reconstruction with 3D supervision Many deep learning based 3D shape reconstruction methods in the literature require the 3D model ground truth to be observed during training [11, 14, 15, 39]. Girdhar et al. [14] propose to learn a joint embedding for both 3D shapes and 2D images with full supervision. Likewise, methods such as Pixel2Mesh [39] and Mesh R-CNN [15] are trained to reconstruct mesh vertices by minimizing the error with respect to the ground truth 3D model. Methods such as O-Net [29] predict if randomly sampled 3D points are inside or outside the 3D models instead of directly approximating their vertices. These methods obtain high reconstruction accuracy. Unfortunately, the cost of acquiring 3D ground truth annotations is high, which motivates research for alternative approaches with weaker supervision such as 2D images and viewpoints.

Reconstruction with Viewpoint Supervision More recently, some works avoid 3D supervision by taking advantage of differentiable renderers [7, 21, 27] with either multiple views, or known ground truth camera poses. Yan et al. [41] leverage perspective transformations and the ground truth viewpoints in order to recover the 3D objects corresponding to the 2D images without 3D ground truth information. Similarly, Gwak et al. [19] leverage 2D perspective projections while constraining the reconstructed 3D shapes to a manifold of 3D unlabeled shapes of real-world objects.
Semi- and Self-Supervised Learning  Leveraging unlabeled data for 3D reconstruction is largely unexplored. Liao et al. [37] introduce a consistency loss that, given two images of an object instance with two different poses, enforces that the shape predicted for one pose is consistent with the depth/mask observation of the second image. Although these works reduce the need for 3D annotations, they still require full supervision with a weaker form of label such as viewpoints. In this work, we propose to automatically approximate viewpoints from unlabeled data thus reducing the amount of required annotations.

3. Methodology

In this paper we propose SSR, a framework for semi-supervised 3D object reconstruction from 2D images. In order to reduce the amount viewpoint supervision, we propose to train a Siamese network (SiamVP) to infer the viewpoint for images for which the viewpoint is unknown. SiamVP takes two images as input and identifies whether they are acquired from the same viewpoint (Figure 3). This network is then used to predict viewpoint pseudo labels for the unlabeled set by comparing them with images with a known viewpoint. These images and the inferred viewpoints are used to train a soft rasterizer network (Figure 1). In the next sections we describe the training data, the pipeline that involves optimizing the soft rasterizer network, the Siamese viewpoint predictor and the training and validation procedure.

3.1. Training Data

We consider the standard semi-supervised learning setting [5], where there is a set of labeled images \( X_L \) whose labels are the viewpoints and a set of unlabeled images \( X_U \) with unknown viewpoint. We assume that the same objects can appear in multiple images under different viewpoints both in \( X_L \) and \( X_U \) and that we know the sets of images that correspond to each object. For each image we have the foreground mask for the object of interest, in some cases like in ShapeNet the input image is also considered the foreground mask. These assumptions are common for this problem setup [24, 37].

3.2. Training Pipeline

As in Figure 1, training consists of optimizing both SiamVP and SoftRas. We describe each of the two components in the following sections.

SoftRas. We train a model for 2D to 3D reconstruction using the procedure described by Liu et al. [27]. First, a 2D image \( I_s \) is provided as input to the SoftRas model. The model consists of an encoder and a decoder and outputs a 3D mesh \( M \) composed of vertices and faces. That 3D representation is projected into a silhouette \( \hat{I}_s \) with the viewpoint associated with that image using a soft rasterizer. Our loss consists of the following two terms: \( L_s + L_g \). The \( L_s \) term corresponds to the silhouette loss:

\[
L_s = 1 - \frac{||\hat{I}_s * I_s||_1}{||\hat{I}_s + I_s - \hat{I}_s * I_s||_1},
\]

where * and + are the element-wise product and sum operators, respectively. \( L_g \) is the geometric loss that regularizes the Laplacian of the shape. Note that this model is amenable to any encoder-decoder architectures. Its key component is the soft rasterizer function that allows the gradients to flow from the IoU loss computed in Eq. 1 to the parameters of the encoder and decoder.

SiamVP. At each iteration SiamVP takes pairs of images, where some of them correspond to the same viewpoint and
others do not (Figure 3). Pairs of images of the same viewpoint are sampled from different objects to avoid the model optimizing a trivial loss. Since most pairs of images have different viewpoints, we sample a balanced set of pairs from images with same and different viewpoints and also mine hard negatives and positives to ensure a stable loss optimization. Hard negatives are those pairs that have dissimilar viewpoints and have the highest predicted probability of being the same, whereas hard positives are those pairs that have the same viewpoints but have the lowest predicted probability of being the same. This procedure is also shown in Step 4 of Algorithm 1.

Thus given two images $I_i$ and $I_j$ and an indicator function $s_{i,j}$ that is 1 when the viewpoint of $I_i$ is the same as the viewpoint of $I_j$ and 0 otherwise, we minimize the following loss function:

$$
\mathcal{L}_{v(i_i, i_j)} = -s_{i,j} \log(P_v) - (1 - s_{i,j}) \log(1 - P_v).
$$

(2)

In the case where only few examples have been labeled, it is important to augment the dataset to avoid overfitting. Between pairs of images with same viewpoint, we also augment the pair with identical random rotation between 0 and 360 degrees and incorporate their similarity probability into the loss in Eq. 2.

**Pseudo-Labeling.** While SiamVP is being trained, we add the viewpoint labels to the unlabeled set of $N_u$ examples as in Step 6 of Algorithm 3.2. The "siam_vp" predict function computes the viewpoint of an unlabeled image as follows. First, from the labeled set we randomly select an image from each viewpoint as reference and compare it with each unlabeled example using SiamVP to get a similarity matrix $S \in \mathbb{R}^{N_u \times N_v}$ where $N_v$ is the number of unique viewpoints we have in the training set. For each row $S_i$ we select the viewpoint $v_j$ with the highest probability $S_{ij}$. We also compute $\hat{S}$ which is the similarity matrix between the labeled and unlabeled examples after they undergo the same rotation. Like with $S$, we also select the viewpoint $\hat{v}_j$ with the highest probability $\hat{S}_{ij}$ for each row $i$. If $S_{ij}$ and $\hat{S}_{ij}$ are both higher than 0.5 and $v_j = \hat{v}_j$ then we select $v_j$ to be the label of unlabeled image $I_i$ otherwise we skip labeling the image. In our experiments, we observe that this approach tends to label a small portion of the unlabeled set each time this procedure is executed, and the accuracy between these predicted viewpoints and the groundtruth is often high.

**The Full Cycle.** Every $z$ iterations of training SiamVP and Softras, we execute the pseudo-labeling procedure from the previous paragraph to get more unlabeled examples labeled with viewpoints. In the first cycle where none of the unlabeled examples have been pseudo labeled, we train Softras using its standard procedure. In the subsequent cycles, each batch that Softras receives consists of half labeled examples, and half pseudo-labeled examples which are trained collectively using the Softras standard loss term which is shown in Step 3 in Algorithm 3.2.

**Inference.** At test time, SiamVP is discarded and the trained Softras takes as an input a single unseen 2D image of an object and outputs the 3D mesh of that object (Figure 3).
Table 1. Number of objects for each of the 13 ShapeNet Categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>Number of Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plane</td>
<td>4045</td>
</tr>
<tr>
<td>Bench</td>
<td>1816</td>
</tr>
<tr>
<td>Cabinet</td>
<td>1572</td>
</tr>
<tr>
<td>Car</td>
<td>7496</td>
</tr>
<tr>
<td>Chair</td>
<td>6778</td>
</tr>
<tr>
<td>Display</td>
<td>1095</td>
</tr>
<tr>
<td>Lamp</td>
<td>2318</td>
</tr>
<tr>
<td>Speaker</td>
<td>1618</td>
</tr>
<tr>
<td>Rifle</td>
<td>2372</td>
</tr>
<tr>
<td>Sofa</td>
<td>3173</td>
</tr>
<tr>
<td>Table</td>
<td>8509</td>
</tr>
<tr>
<td>Phone</td>
<td>1052</td>
</tr>
<tr>
<td>Vessel</td>
<td>1939</td>
</tr>
</tbody>
</table>

Table 2. IoU results for various objects of ShapeNet, where \( N \) is the number of labeled examples.

<table>
<thead>
<tr>
<th>Method</th>
<th>( N )</th>
<th>Plane</th>
<th>Bench</th>
<th>Cabinet</th>
<th>Car</th>
<th>Chair</th>
<th>Display</th>
<th>Lamp</th>
<th>Speaker</th>
<th>Rifle</th>
<th>Sofa</th>
<th>Table</th>
<th>Phone</th>
<th>Vessel</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVC</td>
<td>0</td>
<td>0.38</td>
<td>-</td>
<td>0.39</td>
<td>0.6</td>
<td>0.26</td>
<td>0.33</td>
<td>0.17</td>
<td>0.38</td>
<td>0.44</td>
<td>0.38</td>
<td>0.17</td>
<td>0.57</td>
<td>0.28</td>
<td>0.36</td>
</tr>
<tr>
<td>SoftRas</td>
<td>2</td>
<td>0.42</td>
<td>0.24</td>
<td>0.39</td>
<td>0.6</td>
<td>0.26</td>
<td>0.33</td>
<td>0.17</td>
<td>0.38</td>
<td>0.44</td>
<td>0.38</td>
<td>0.17</td>
<td>0.57</td>
<td>0.28</td>
<td>0.36</td>
</tr>
<tr>
<td>SSR (ours)</td>
<td>2</td>
<td>0.55</td>
<td>0.44</td>
<td>0.48</td>
<td>0.75</td>
<td>0.37</td>
<td>0.5</td>
<td>0.27</td>
<td>0.45</td>
<td>0.63</td>
<td>0.53</td>
<td>0.27</td>
<td>0.71</td>
<td>0.53</td>
<td>0.50</td>
</tr>
<tr>
<td>SoftRas</td>
<td>5</td>
<td>0.43</td>
<td>0.27</td>
<td>0.42</td>
<td>0.64</td>
<td>0.29</td>
<td>0.34</td>
<td>0.19</td>
<td>0.45</td>
<td>0.43</td>
<td>0.47</td>
<td>0.2</td>
<td>0.60</td>
<td>0.42</td>
<td>0.40</td>
</tr>
<tr>
<td>SSR (ours)</td>
<td>5</td>
<td>0.59</td>
<td>0.44</td>
<td>0.5</td>
<td>0.75</td>
<td>0.49</td>
<td>0.55</td>
<td>0.33</td>
<td>0.47</td>
<td>0.65</td>
<td>0.66</td>
<td>0.35</td>
<td>0.78</td>
<td>0.56</td>
<td>0.55</td>
</tr>
<tr>
<td>SoftRas</td>
<td>20</td>
<td>0.47</td>
<td>0.34</td>
<td>0.53</td>
<td>0.68</td>
<td>0.33</td>
<td>0.44</td>
<td>0.26</td>
<td>0.51</td>
<td>0.52</td>
<td>0.55</td>
<td>0.31</td>
<td>0.67</td>
<td>0.47</td>
<td>0.47</td>
</tr>
<tr>
<td>SSR (ours)</td>
<td>20</td>
<td>0.63</td>
<td>0.47</td>
<td>0.65</td>
<td>0.76</td>
<td>0.51</td>
<td>0.57</td>
<td>0.38</td>
<td>0.62</td>
<td>0.66</td>
<td>0.65</td>
<td>0.43</td>
<td>0.78</td>
<td>0.55</td>
<td>0.59</td>
</tr>
</tbody>
</table>

SoftRas All | 0.64  | 0.51  | 0.71  | 0.77  | 0.53  | 0.62  | 0.46    | 0.67 | 0.68    | 0.69  | 0.45 | 0.79  | 0.60  | 0.62   |

Figure 4. Results Across Pseudo-Labeling Cycles. The left plot depicts the percentage of unlabeled examples that has been pseudo-labeled by SiamVP. The center plot shows the accuracy of the pseudo labels across epochs. The right plot shows the 3D IoU on the ShapeNet’s test set.

4. Experiments

We evaluate the effectiveness of SSR for 3D reconstruction and viewpoint estimation. We first report the performance of SSR when trained with different numbers of examples and compare it with different baselines with different amounts of supervision. Then we study the evolution of SSR’s performance across pseudo-labeling cycles and provide ablations to determine the influence of using augmentations and metric learning.

4.1. Setup

We experiment on ShapeNet v2 dataset [4] which contains a rich collection of 3D computer-aided design (CAD) models of different classes of objects. Shapenet is widely used in recent research works related to 2D/3D data. We focus on 13 standard categories of ShapeNet which are Plane, Bench, Cabinet, Car, Chair, Display, Lamp, Speaker, Rifle, Sofa, Table, Phone, Vessel as they offer enough objects to build a diversified image dataset and they were previously used in a similar setup like Liu et al. [27]. The number of objects in each class is shown in Table 1. We benchmark with renderings provided by Kato et al. [21], where the 3D CAD object models were rendered at 24 uniform viewpoints to 64 × 64 images. We split the dataset into training, validation and testing sets accounting for 70, 10 and 20 percent of the whole dataset respectively, following Yan et al. [41]. To simulate a semi-supervised setup, we further split the training set by randomly selecting a subset of the data to act as the labeled set, and the rest acting as unlabeled. We adjust the number of labeled samples in our experiments to show the effect of varying degrees of supervision. The splits are made on a object basis, that is, the different views from the same 3D model are either all labeled or all unlabeled.

We evaluate the 3D reconstruction accuracy by the 3D Intersection of Union (3D IoU) between the reconstructed images from every of the 24 viewpoint and ground truth 3D voxels of the test objects.
### Algorithm 1: PyTorch-style pseudocode for SSR

#### 4.2. Implementation Details

We use the same structure and hyper-parameters as [27] for mesh generation. Our network is optimized using Adam with $\alpha = 1 \cdot 10^{-4}$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We used a batch size of 64 and implemented our pipeline using PyTorch by building on the code found here.\(^1\) For the SiamVP architecture, we used the Siamese network defined by Koch et al. [22] with weights initialized using Kaiming Initialization [20]. We observed that this network attained better generalization than if we used the encoder architecture from Liu et al. [27] which is more optimized towards constructing 3D meshes. We used a batch size of 32 for SiamVP and optimized the network using Adam with learning rate $10^{-4}$.

These hyper-parameters were coarsely optimized using the validation set but better results would be likely achieved with more tuning on the validation set which would only strengthen the contributions made by this work. Note that we use the validation set for early stopping and we report the test score achieved by the model that achieved the best result for the validation set in the 1000 epochs of training.

#### 4.3. Experimental Results

**Comparison with the baseline** We compare the proposed SSR (SoftRas +SiamVP) against SoftRas for the cases where only few objects have been labeled. We report the results in Table 2. We observe that by labeling only 2, 5, or 20 ShapeNet objects for each class with viewpoints, SSR consistently achieves a significantly better IoU.
test score than SoftRas. SSR uses the procedures described in Section 3 which leverages the unlabeled set by pseudo labeling it with SiamVP. In Figure 5 we observe that SoftRas tends to overfit on the labeled set and fails to adapt to new objects. On the other hand, SSR's reconstructions are more similar to their original counterpart. This result suggests that SiamVP can accurately predict the viewpoints of the unlabeled set which can be effectively used by SoftRas.

Comparison with an unsupervised method MVC [37] is an unsupervised method that can learn camera parameters from objects given multiple views for each of them. It does not use any viewpoint labels and it does not use differentiable rendering, thus it is a significantly different approach for 3D reconstruction than SoftRas. Its score is reported in Table 2. Although MVC achieved good results considering that it used no viewpoint labels, SSR makes a significant improvement with only 2 objects being labeled, especially for the categories Plane, and Car. Unfortunately, MVC [37] does not report results for the remaining classes in order to make further comparison. The results however indicate that it is worth labeling extra objects and to use SSR to get a major increase in performance.

Comparison across number of labeled objects We compare the performance achieved from labeling 2, 5, or 20 objects from the training set in Table 2. We observe that SSR did not gain statistically significant improvement for classes like Car, Vessel, and Rifle. Our hypothesis is that many of the objects look similar in these classes, so labeling few extra objects might not provide rich information that SoftRas could use for better 3D reconstruction. On the other hand, for classes where objects are more diverse like Plane, Chair, Lamp, and Sofa, SSR achieved big improvement with 5 labeled objects than with 2 labeled objects. We also see that there is a big boost in the mean iou test score as the number of labeled objects increase, which suggests that SiamVP can still be effective with more training labels.

Comparison with the upper bound In Table 2 we also report results for SoftRas trained on the full labeled set. We observe that the performance gap between it and SSR (with 20 objects labeled) is small for most classes like Plane, Car, Display, Rifle. Note that 20 objects is a small number compared to the full training set size which are shown in Table 1. For the Plane class, 20 only makes 0.7% of the total number of Plane objects in the training set. This result indicates that for some classes few labels might be sufficient to get good results while for other classes, like Cabinet and Speaker, we would need more labels. This observation leads to an important future direction that could involve active learning to identify which objects are best to label for maximum information gain.

Comparison with a viewpoint classifier We also conducted an experiment where we trained a viewpoint classifier using the same encoder as SiamVP that takes as input ShapeNet images and outputs the viewpoint of the image from a large list of hypothesis viewpoints. We observed a severe overfitting of the model on the labeled set and the results were worse than SoftRas that was trained only on the labeled set. Therefore, our metric learning approach with SiamVP is significantly more robust for classifying viewpoints than with a viewpoint classifier when the labeled set is small.

Results Across Pseudo-Labeling Cycles We investigate our model’s performance as we grow the number of unlabeled examples that get incorporated to the training set with SiamVP’s pseudo labels. Pseudo-labeling cycles occur every 2 epochs, where every epoch consists of 200 training iterations. In Figure 4 we observe that the first set of labeled examples induce high accuracy gains on viewpoint prediction. However, as more viewpoints get labeled we observe some decrease in the viewpoint accuracy. For some shapes like Car, the accuracy drops from 90% to 88% suggesting that SiamVP might start to overfit as the training progresses.

On the other hand, we see that the accuracy plateaus above 80% which indicates that SiamVP is robust to pseudo-labeling noise. Further, we see that for Plane objects the ratio of pseudo-labeled samples is almost 100% whereas for Chair it is 60%, suggesting that for some chair objects, SiamVP’s predicted viewpoint probability was low.

Note that SiamVP is training alongside SoftRas so at each pseudo-labeling cycle there is a chance that SiamVP labels a different set of unlabeled examples. We observed that this approach resulted in more accurate pseudo-labels across all object sets.

We also observe that the highest increase in the test IoU occurs in the first 20 epochs with a slight increase later in training. As mentioned in the implementation details, in the rest of the experiments we only report the test IoU at the end of their respective plot lines which correspond to the best validation model.

The importance of augmentation. Described in Section 3, the goal of augmentation is to help SiamVP generalize better as there is a limited number of viewpoints in the labeled set. In Table 3 we compare the results with and without the augmentation for 4 ShapeNet classes where the number of labeled objects is 2. We observe that there is

<table>
<thead>
<tr>
<th>Method</th>
<th>Plane</th>
<th>Bench</th>
<th>Cabinet</th>
<th>Car</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSR w/o augmentation</td>
<td>0.45</td>
<td>0.30</td>
<td>0.43</td>
<td>0.64</td>
</tr>
<tr>
<td>SSR (ours)</td>
<td>0.55</td>
<td>0.44</td>
<td>0.48</td>
<td>0.75</td>
</tr>
</tbody>
</table>
consistent improvement in the results when augmentation is added as it helps the model learn the 3D mesh while being invariant on the viewpoint of the 2D images. Note that for this approach we only used the rotation augmentation but other types of augmentation such as scaling, and translation can be used as well. Another augmentation technique to investigate is using a pretrained SoftRas to perform projections under novel viewpoints and use those projections as input to train a SiamVP.

5. Conclusion

We propose SSR, a semi-supervised soft rasterizer method that leverages unlabeled examples by using accurate pseudo labels. SSR consists of a viewpoint estimation component and a soft rasterization component. The viewpoint estimation is performed by a Siamese network called SiamVP that outputs whether two images have the same viewpoint. SiamVP slowly pseudo-labels images by carefully choosing only the most confident viewpoint predictions at each cycle. These viewpoint pseudo-labeling procedure leads to a significant boost in the performance of SoftRas for reconstructing single 2D images to their 3D representation. Future directions for this work would include adopting this method for predicting viewpoints for objects in the wild where the environment is more challenging and less constrained than the ShapeNet dataset.
References


