ShapeClipper: Scalable 3D Shape Learning from Single-View Images via Geometric and CLIP-based Consistency

Zixuan Huang\textsuperscript{1} Varun Jampani\textsuperscript{2} Anh Thai\textsuperscript{1} Yuanzhen Li\textsuperscript{2} Stefan Stojanov\textsuperscript{1} James M. Rehg\textsuperscript{1}
\textsuperscript{1}Georgia Institute of Technology, \textsuperscript{2}Google Research

Figure 1. We propose a method that reconstructs 3D object shape from single view real-world images. Our method learns high-quality reconstruction through \textit{single-view supervision without known viewpoint} and can reconstruct shapes of various objects.

Abstract

We present ShapeClipper, a novel method that reconstructs 3D object shapes from real-world single-view RGB images. Instead of relying on laborious 3D, multi-view or camera pose annotation, ShapeClipper learns shape reconstruction from a set of single-view segmented images. The key idea is to facilitate shape learning via CLIP-based shape consistency, where we encourage objects with similar CLIP encodings to share similar shapes. We also leverage off-the-shelf normals as an additional geometric constraint so the model can learn better bottom-up reasoning of detailed surface geometry. These two novel consistency constraints, when used to regularize our model, improve its ability to learn both global shape structure and local geometric details. We evaluate our method over three challenging real-world datasets, Pix3D, Pascal3D+, and OpenImages, where we achieve superior performance over state-of-the-art methods.\textsuperscript{1}

1. Introduction

How can we learn 3D shape reconstruction from real-world images in a scalable way? Recent works achieved impressive results via learning-based approaches either with 3D \cite{4, 9, 11, 28, 36, 41, 43–45, 48, 49, 53} or multi-view supervision \cite{16, 19, 23, 25, 32, 42, 50, 51}. However, such supervised techniques cannot be easily applied to real-world scenarios, because it is expensive to obtain 3D or multi-view supervision at a large scale. To address this limitation, recent works relax the requirement for 3D or multi-view supervision \cite{1, 8, 10, 14, 15, 17, 18, 22, 24, 29, 31, 38, 46, 55, 57}. These works only require single-view self-supervision, with some additionally using expensive viewpoint annotations \cite{17, 18, 24, 38, 57}. Despite this significant progress, most methods still suffer from two major limitations: 1) Incorrect top-down reasoning, where the model only explains the input view but does not accurately reconstruct the full 3D object shape; 2) Failed bottom-up reasoning, where the model cannot capture low-level geometric details such as concavities. How can we address these limitations while also remaining scalable to a wide range of object types?

To improve top-down reasoning, our inspiration comes...
from the recent success of large-scale image-text modeling. The most successful image-text models such as CLIP [33] are trained on a vast corpus of captioned images and are able to extract fine-grained semantic features that correlate well with the language descriptions. CLIP further demonstrates a great generalization ability to images across various domains. Can we leverage such a powerful and generalizable model to learn 3D reconstruction in a real-world scenario?

We observe that natural language descriptions of images often contain geometry-related information (e.g. a round speaker, a long bench) and many nouns by themselves have characteristic shape properties (e.g. “desks” usually have four legs, and “benches” normally include a flat surface). Motivated by this intrinsic connection between object shapes and language-based semantics, we examine the latent space of CLIP’s visual encoder. In our study, we find (via k-NN queries) that objects with similar CLIP embeddings usually share similar images (see Fig. 2 for an example). Another key characteristic we identify with CLIP embeddings is that they have some robustness to viewpoint changes, meaning that changes in viewpoint generally do not produce drastic changes in CLIP embeddings.

Inspired by these findings, we propose to learn shapes using a semantic-based shape consistency (SSC) constraint using CLIP. Specifically, we use CLIP’s semantic encodings as guidance to form pseudo multi-view image sets. For each image in the training set, we extract its CLIP embedding and find images with the most similar semantics across the training set. We then leverage these retrieved images as additional supervision to the input view, as illustrated in Fig. 3. This approach greatly benefits global shape understanding, because each predicted shape is required to simultaneously explain a set of images instead of only explaining the single input image.

On the other hand, we address the limitation of poor bottom-up geometric reasoning by constraining the surface normals of the predicted shapes. Common failure cases include noisy surface reconstruction and failed concavity modeling, which are extremely hard to learn even with multi-view supervision. Inspired by the recent success of large-scale 2.5D estimation that generalizes to various scenes [7, 34, 35], we propose to use off-the-shelf surface normals as additional geometric supervision for our task. However, unlike scenes, off-the-shelf normals for object-centric images are much noisier due to occlusion, truncation, and domain gaps. To mitigate this issue, we introduce a noise-tolerant optimization process via outlier dropout, which stabilizes the training and improves the overall reconstruction performance.

Overall, our contributions are threefold:

- We propose a novel CLIP-based shape consistency regularization that greatly facilitates the top-down understanding of object shapes.
- We successfully leverage off-the-shelf geometry cues to improve single-view object shape reconstruction for the first time and handle noise effectively.
- We perform extensive experiments across 3 different real-world datasets and demonstrate state-of-the-art performance.

2. Related Work

There has been an emerging interest in 3D object shape reconstruction from images via learning-based approaches. Our work focuses on learning single-view shape reconstruction with limited supervision on real-world approaches, where the training set only contains a single view per object instance. We briefly survey the relevant literature on single image shape reconstruction using both fully-supervised and weakly-supervised approaches.

Single-View Supervision. Most closely related to this paper are works that learn 3D shape reconstruction through
supervision from single-view images [1, 10, 13–15, 17, 18, 22, 24, 29, 31, 46, 55, 57]. These works can be organized as in Tab. 1 and largely differ in their choice of 1) shape representation (e.g. implicit SDF vs explicit mesh); 2) known vs. unknown viewpoint supervision; 3) large-scale evaluation on various real-world objects. We are one of the earliest works to demonstrate the feasibility of single-view learning of an implicit SDF representation upon diverse real-world images under unknown viewpoints.

Within this body of work, SSMP [55], Cat3D [15], and SS3D [1] are the most closely related ones given their large-scale evaluation, which we describe in details below.

SSMP [55] is the earliest work that shows success of shape learning via only single-view supervision on large-scale real-world data. A key property of this method is adversarial regularization during training, which can make training unstable. Thus it is hard for SSMP to learn reconstruction on categories with complex shapes or textures. In contrast, our method leverages the SSC and geometric constraints which are more stable and result in superior performance over SSMP across various objects.

Similar to our method, Cat3D [15] explores semantic regularization for implicit shape learning. In contrast, their semantic regularization is based on category labels, which fails on categories with significant intra-category shape variations. Moreover, Cat3D relies on unstable adversarial regularization which has similar drawbacks as SSMP [55] and is only successful on a few real-world categories.

SS3D [1] proposes a 3-step learning pipeline for scalable learning of shapes, which includes synthetic data (e.g. ShapeNet [3]) pretraining. This step plays a crucial role as it provides the necessary initialization for the camera multiplex optimization. Unlike SS3D, synthetic pretraining is not a hard constraint for our method—we demonstrate success on Pix3D [39] without any synthetic pretraining. On the other hand, SS3D does not explore the usage of semantic and geometric consistency. As a result, our model captures both global structures and local surfaces more accurately than SS3D and outperforms SS3D quantitatively.

### Shape Supervision

Instead of using image supervision, many prior works use explicit 3D geometric supervision and achieve great reconstruction results [4, 9, 11, 28, 36, 41, 43–45, 48, 49, 53]. Nevertheless, the assumption of 3D supervision is not yet practical on a large scale. To make the learning more scalable, subsequent works leverage multi-view images as supervision and employ differentiable rendering as the core technique. Specifically, differentiable rendering allows images and masks to be rendered from 3D assets differentiably and thus the multi-view reprojection loss can effectively carve the reconstructed shape. These methods can be classified based on their representation of shape, including voxels [42, 50, 51], pointclouds [16, 23], meshes [19, 25] and implicit representations [32]. Compared to these works, a major benefit of our method is scalability, as our model can be trained using single-view real-world images.

### 3. Method

In this section, we first present an overview of our model in Sec. 3.1, and then introduce our proposed SSC and geometric constraints in Sec. 3.2 and Sec. 3.3. Finally, we present implementation details in Sec. 3.4.

#### 3.1. Overview

Given a collection of $n$ images segmented with foreground masks $\{I_i \in \mathbb{R}^{h \times w \times 3}, M_i \in \mathbb{R}^{h \times w \times 1}\}_{i=1}^n$, we aim to learn a single-view 3D reconstruction model without relying on 3D, viewpoint, or multi-view supervision of these images. The shape representation of our model is an implicit SDF function, represented by a multi-layer perceptron (MLP) conditioned on the input image. Specifically, our model consists of four submodules (see Fig. 4 for an overview) as described below.

**Image encoder.** The image encoder $E$ takes a segmented image $I \in \mathbb{R}^{h \times w \times 3}$ as input and infers the shape latent code $s \in \mathbb{R}^l$ and the texture latent code $t \in \mathbb{R}^l$. These two codes encode the necessary information to reconstruct the shape and texture field respectively.

**Shape and texture reconstructor.** Our model represents shape and texture reconstruction with two MLPs, $f_S : \mathbb{R}^3 \rightarrow \mathbb{R}$ and $f_T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, which predict signed distance function (SDF) and RGB values of queried signed distance respectively. The MLPs are conditioned on the latent codes, with a similar design to VolSDF [54]. Specifically, the shape MLP $f_S$ is conditioned on $s$ and the texture MLP $f_T$ is conditioned on $t$. 

![Network overview](image-url)

**Fig. 4. Network overview.** Given the input image, the encoder $E$ infers the shape latent code $s$ and texture latent code $t$. By conditioning these two codes upon shape MLP $f_S$ and texture MLP $f_T$, we obtain the shape and texture reconstruction of the input object. On the other hand, the viewpoint estimator $V$ estimates the input viewpoint $v$. The differentiable volume renderer $R$ then renders shape and texture fields under the estimated viewpoint, so that we can compute the reconstruction loss $\mathcal{L}_1$ and $\mathcal{L}_M$. We further leverage our SSC and geometric constraints, $\mathcal{L}_{SSC}$ and $\mathcal{L}_{N}$, to effectively harness the shape learning.
Table 1. **Single-view supervised methods for object shape reconstruction.** M: mesh, V: voxel, P: pointcloud, D: depth, O: occupancy function, S: signed distance function, Diverse R-Res.: real-world results on diverse categories.

<table>
<thead>
<tr>
<th>Model</th>
<th>[17]</th>
<th>[18]</th>
<th>[24]</th>
<th>[38]</th>
<th>[57]</th>
<th>[10]</th>
<th>[22]</th>
<th>[14]</th>
<th>[13]</th>
<th>[31]</th>
<th>[46]</th>
<th>[29]</th>
<th>[15]</th>
<th>[55]</th>
<th>[1] Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Rep.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Viewpoint Free</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Diverse R-Res.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Viewpoint estimator.** The viewpoint estimator $V$ estimates the viewpoint of the input image with respect to the shape reconstruction. Following [2, 15], we represent the viewpoint with the trigonometric functions of Euler angles, i.e. $v = [\cos \gamma, \sin \gamma]$ where $\gamma$ denotes the three Euler angles.

**Differentiable renderer.** We use a volume renderer $R$ to render the reconstructed SDF and texture fields following VolSDF [54]. In the renderer, the SDF field is first converted into densities, and then the densities are used together with the texture field to render the RGB and mask in an accumulative way (via ray-marching). We refer the readers to VolSDF [54] for more details, with the exception that we use uniform sampling instead of error-bound based sampling. Formally, we denote the renderer as a functional, $R(f_S, f_T, v)$, which maps the implicit functions and the viewpoint into image $I$, mask $M$, and surface normal $N$.

**Reprojection loss.** One of the major learning signals of our model comes from the reprojection loss that compares input images with reconstructed images. This can be achieved via the differentiable renderer. Our model first infers shape, texture, and viewpoint of the object from the input image. The renderer can then render them into an image reconstruction, which will be matched to the input.

Specifically, we can denote the RGB and the mask reprojection loss for each image as follows:

$$L_I = \|I - \hat{I}\|^2_2, \quad L_M = 1 - IOU(M, \hat{M}),$$

$$IOU(M, \hat{M}) = \frac{\sum_{p} M^p \cdot \hat{M}^p}{\sum_{p} M^p + \hat{M}^p - M \cdot \hat{M}^p}.$$  

Here $M^p$ denotes the mask value at pixel $p$.

**Facilitating shape learning.** When we do not have direct viewpoint or shape supervision, simply minimizing the reprojection loss almost always leads to degeneration. There are two major issues: 1) incorrect top-down reasoning, where shapes can only explain the input view; 2) wrong bottom-up reasoning, examples include the inability to infer concavity or noisy surface reconstruction. To mitigate these issues, we propose the semantic and geometric consistency constraints that effectively facilitate the shape learning.

3.2. Semantic Constraint

**Preliminary findings about CLIP.** To leverage CLIP for shape regularization, our main hypothesis is that objects with similar CLIP encodings share similar shapes. To verify this hypothesis, we perform a study using the large-scale fine-grained CompCars [52] dataset. This dataset contains more than 136K images of 163 car makes with 1716 car models. We perform CLIP inference on this dataset and compute 5-nearest neighbor for each sample based on the CLIP embeddings. By iterating over each neighbor of all query images, we calculate the percentage of neighbors that match their query images’ model (same car model usually shares quite similar shapes). In our experiment, CLIP is able to find the exact same car models for 51.2% of all the neighbors (on average 2.6 out of 5 neighbors belong to the query images’ model). We believe this is a promising finding given 1) the large number of images and models in CompCars and 2) the fact that different models can still have similar shapes, so the percentage of shape matches can be higher than exact model matches. As a comparison, the percentage of model matches for ImageNet-pretrained ViT [6] is only 27.8%. This study verifies our hypothesis and enables us to design our Semantic-based Shape Consistency (SSC) constraint based on CLIP.

**Semantic-based Shape Consistency.** The key idea of SSC is to pull instances with similar CLIP embeddings together, so that a single shape reconstruction can receive supervision from all these instances. In our experiments, we find CLIP encodings have some robustness to viewpoint change (see supplement for more details). Therefore it enables us to find additional pseudo views for many objects, which significantly facilitate the learning of better top-down reasoning.

We first form the per-instance clusters by performing K-nearest neighbors with CLIP encodings. Formally, given our training set $\{I_i\}$, we extract the CLIP encoding for each image, denoted as $\{c_i\}$. We calculate the cosine similarity of all pairs of encodings. With such similarity measurement, we can query the K-nearest neighbors for any specific input encoding $\{c_i\}$ and identify images and masks of these neighbors.

We then use these semantic neighbors to supervise the shape reconstruction as in Fig. 5. The high-level idea is that 1) the shape reconstructed for the input should explain neighbors’ masks and normals, and 2) when combining input’s shape with neighbor’s texture, we should be able to render the neighbor image.

Formally, for input $I$, we denote its shape latent code and shape MLP as $s$ and $f_s$. Meanwhile, we sample an image and its mask $I_k$ and $M_k$ from the semantic neighbor set $\{I_k, M_k\}_{k=1}^K$. The encoder $E$ then predicts the latent texture code $t_k$ for $I_k$, which is used to generate its texture.
where $R$ refers to the rotation matrix derived from the estimated viewpoint and $\cos$ denotes the cosine similarity. We set $\beta = 5$ across all the experiments.

This geometric loss is calculated at the pixel level and averaged over a minibatch. However, unlike scene reconstruction [56], off-the-shelf normals can be noisy for object-centric images due to inaccurate masks and domain gaps. As a result, naively using off-the-shelf normals results in training instability. Inspired by online hard example mining [37], we propose to dropout off-the-shelf normals that are likely to be outliers via batchwise ranking. Specifically, we sort the normal loss $L_N$ within the current minibatch and exclude a fixed percentage of high-loss pixels from the final loss aggregation. We find that this strategy stabilizes the training and improves the reconstruction quality overall.

Finally, we can combine the geometric constraint with the semantic constraint by having a SSC normal loss, $L_{SSCN}$. This can be calculated similarly to $L_N$, the only difference is that we replace the input off-the-shelf normals and rotations with the semantic neighbor’s as in Eq. (3).

### 3.4. Implementation Details

**Architecture.** The image encoder we use is a ResNet34 [12], which projects the input image into two 64-d latent vectors representing shape and texture. We use lightweight MLPs to represent the SDF and texture fields, where the shape MLP has 5 hidden layers of 64 neurons and the texture MLP has 3 hidden layers of 64 neurons. The 3D coordinates are positionally encoded [56]. The 3D coordinates are positionally encoded [56].

Formally, we denote our surface normal estimation as $\hat{N} \in \mathbb{R}^{h \times w \times 3}$ and the off-the-shelf unit normal as $N \in \mathbb{R}^{h \times w \times 3}$. The estimated normal is calculated as the normalized gradient of the density and aggregated via volume rendering similar to MonoSDF [56]. Unlike the setup in MonoSDF, our normal estimation lies in the object-centric canonical frame instead of the view-centric frame. Therefore, we use our estimated viewpoint to rotate the off-the-shelf normal $N$ to be in the same canonical frame as $\hat{N}$. In addition to aligning the coordinate frames, this approach enables the viewpoint estimator to receive additional training signals from local geometry alignment, which is a significant benefit that naive approaches like the closed-form rotation alignment cannot provide. After the rotation, we can then match the normals following [7]:

$$L_N = \beta \cdot \|RN - \hat{N}\|_1 - \cos(RN, \hat{N}),$$  (4)

$$L_{SSC} = L_{SSC_i} + \lambda_1L_{SSCM} + \lambda_2L_{SSCN},$$  (6)

We set $\lambda_1 = 0.5$ and $\lambda_2 = 0.01$ across all datasets.

**Training.** We use the Adam [20] optimizer with a learning rate of 0.0001 and a batch size of 12. We did not use weight decay or learning rate scheduling. Instead of using all pixels at once, we sample 512 rays to perform ray-marching for each image. Our model is trained on a single NVIDIA GTX TITAN Xp for 200 to 400 epochs depending on the dataset size, which usually takes 1 to 3 days to train. Following SS3D [1], we initialize the model by pretraining on
ShapeNet for our experiments on Pascal3D+ and OpenImages, where we compare to SS3D. We use the commonly used symmetry constraint [30, 55] and regularize the azimuth with a uniform prior. We further regularize the SDF field with the eikonal loss so the gradient norm is close to 1. For Pix3D we did not use synthetic pretraining, and instead we pretrain the shape to a sphere for a better initialization similar to Cat3D [15].

4. Experiments

We present the findings of applying our method across three real-world datasets in this section, including state-of-the-art comparison and detailed ablations. We first introduce the datasets we use and then describe the evaluation metrics as well as the baselines. Finally, we show detailed experiment results on each dataset.

4.1. Datasets

We evaluate our method across three real-world datasets, including Pix3D [39], Pascal3D+ [47] and OpenImages [21].

Pix3D. Pix3D is a real-world 3D object dataset where each image is annotated with a corresponding object mask, a CAD model, and the input viewpoint. This 3D information is obtained via manual alignment between shapes and images. We use the chair category, which is the dominant category of this dataset. We follow the 70/10/20 split of [15], resulting in 2007, 303, 584 images for training/validating/testing respectively.

Pascal3D+. Pascal3D+ is a real-world 3D object dataset obtained similarly to Pix3D. Compared to Pix3D, this dataset is more challenging because 1) it includes 12 diverse categories with a high-variance viewpoint distribution, and 2) object masks are quite noisy and some objects are occluded. We use all categories in the ImageNet [5] subset of this dataset, with a similar split to [15, 24], resulting in 11317, 11421 images for training and testing respectively.

OpenImages. Unlike Pix3D and Pascal3D+, OpenImages do not contain any 3D annotation, which prevents us from evaluating methods quantitatively. We use 20 diverse categories (more in supplement) to train our model and leverage the occlusion scores from [55] to filter out highly occluded images. Each category includes 1000 to 3000 images, which are split into 90/10 for training and testing.

4.2. Evaluation

To evaluate the implicit shapes, we sample SDF values with a $100^3$ spatial grid and extract the 0-isosurface via Marching Cubes [26]. We further align the coordinate frames between predicted meshes and ground truth (GT) meshes, by transforming all meshes into the view-centric coordinate frame. We also align the scales of these meshes using the size of their projections on the image plane. After these steps, we can sample points from mesh surfaces and calculate Chamfer Distance and F-score as our quantitative metrics following [11, 15, 24, 40, 41].

Chamfer Distance. Following [15], Chamfer Distance (CD) is defined as an average of accuracy and completeness. Given two pointclouds $S_1$ and $S_2$, CD can be written as:

$$d(S_1, S_2) = \frac{1}{2|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x-y\|_2 + \frac{1}{2|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|x-y\|_2$$

(8)

F-score. F-score (FS@d) is a joint measurement of accuracy and completeness with a given threshold $d$. Specifically, precision@d is the percentage of predicted points that have at least one GT neighbor within distance $d$. Similarly, recall@d is the percentage of ground truth points that have at least one neighboring predicted points within distance $d$. FS@d is then calculated as the harmonic mean of precision@d and recall@d. It can be intuitively understood as the percentage of surface that are correctly reconstructed.
Table 2. Quantitative results on Pix3D. Our method performs favorably to baselines and other SOTA methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FS@1↑</th>
<th>FS@5↑</th>
<th>FS@10↑</th>
<th>CD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o $L_{SSC}$</td>
<td>0.0958</td>
<td>0.4309</td>
<td>0.7093</td>
<td>0.749</td>
</tr>
<tr>
<td>w/o normal</td>
<td>0.0815</td>
<td>0.3913</td>
<td>0.6982</td>
<td>0.766</td>
</tr>
<tr>
<td>w/o noise-tol</td>
<td>0.1277</td>
<td>0.5319</td>
<td>0.7861</td>
<td>0.640</td>
</tr>
<tr>
<td>Ours</td>
<td>0.1317</td>
<td>0.5473</td>
<td>0.8002</td>
<td>0.618</td>
</tr>
<tr>
<td>Cat3D [15]</td>
<td>0.0960</td>
<td>0.4410</td>
<td>0.7262</td>
<td>0.679</td>
</tr>
<tr>
<td>SSMP [55]</td>
<td>0.0948</td>
<td>0.4261</td>
<td>0.7168</td>
<td>0.707</td>
</tr>
</tbody>
</table>

4.3. Baselines

We consider three different baselines in this work, including SSMP [55], Cat3D [15] and SS3D [1].

SSMP learns single-view supervised voxel reconstruction via adversarial regularization.\(^2\) We compare our method to SSMP over Pix3D and Pascal3D+.

Cat3D learns multi-class shape reconstruction without using any 3D/viewpoint annotation. It uses a similar implicit SDF representation. We compare our method to Cat3D over Pix3D and Pascal3D+.

SS3D learns single-view supervised implicit shape reconstruction by pretraining on ShapeNet first. They use a per-instance camera multiplex optimization, which is too computationally expensive for us to train their model (based on their paper, even training on a single category-specific model takes 64 V100 days on average). Therefore, we use their publicly available pretrained weights instead and evaluate their method on Pascal3D+ by selecting 11 categories that SS3D has seen during training. Additionally, because SS3D cannot predict viewpoints during inference, we use the brute-force evaluation to evaluate shape reconstruction when comparing our method to SS3D. Specifically, for each instance, we align the scales of the reconstructed shape and the GT shape, and search for the rotation that leads to the lowest Chamfer Distance. We also compare our method to SS3D on OpenImages qualitatively.

4.4. Pix3D

We perform experiments on Pix3D and show quantitative and qualitative results in Tab. 2 and Fig. 7.

Ablation Study. We first analyze the results of ablating the techniques we propose. In Tab. 2, ‘w/o $L_{SSC}$’ refers to our model without the CLIP-based semantic constraint, ‘w/o normal’ refers to our model without the geometric constraint, ‘w/o noise-tol’ refers to our model without outlier dropping in the geometric constraint. Because we do not perform synthetic pretraining on Pix3D, we find the final performance of the baselines sensitive to the initialization. Therefore we average the quantitative results over 5 runs with different random seeds. By comparing the baselines to our final model, we clearly see our semantic and geometric constraints improve the reconstruction performance, and the outlier dropping benefits the shape learning as well. In the qualitative results (Fig. 7), we find our semantic constraint leads to better global structures, while our geometric constraint significantly improves the reconstruction of object surfaces.

SOTA Comparison. Comparing with SOTA methods (Tab. 2, last 3 rows), we see our approach outperforms Cat3D and SSMP significantly. Qualitatively, our approach captures better overall shape topology and local geometric details than Cat3D and SSMP. These results all demonstrate the effectiveness of our proposed method. Note that for comparisons on this dataset there is no synthetic pre-training for any methods.
Table 3. **Quantitative results on Pascal3D+**. Our method performs favorably to baselines and other SOTA methods.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FS@1↑</th>
<th>FS@5↑</th>
<th>FS@10↑</th>
<th>CD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o $L_{SSC}$</td>
<td>0.1363</td>
<td>0.5268</td>
<td>0.7307</td>
<td>0.898</td>
</tr>
<tr>
<td>w/o normal</td>
<td>0.1185</td>
<td>0.4648</td>
<td>0.6712</td>
<td>0.952</td>
</tr>
<tr>
<td>w/o noise-tol</td>
<td><strong>0.1548</strong></td>
<td>0.5875</td>
<td>0.7874</td>
<td>0.707</td>
</tr>
<tr>
<td>Ours</td>
<td>0.1519</td>
<td><strong>0.5914</strong></td>
<td><strong>0.7981</strong></td>
<td><strong>0.693</strong></td>
</tr>
<tr>
<td>Cat3D [15]</td>
<td>0.0858</td>
<td>0.3977</td>
<td>0.6155</td>
<td>1.118</td>
</tr>
<tr>
<td>SSMP [55]</td>
<td>0.1014</td>
<td>0.4366</td>
<td>0.6614</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Table 4. **Additional quantitative comparison to SS3D on Pascal3D+**. Shapes are aligned via brute-force search. Our method performs favorably to SS3D on Pascal3D+.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FS@1↑</th>
<th>FS@5↑</th>
<th>FS@10↑</th>
<th>CD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS3D [1]</td>
<td>0.0533</td>
<td>0.4879</td>
<td>0.7702</td>
<td>0.696</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.0585</strong></td>
<td><strong>0.5388</strong></td>
<td><strong>0.8507</strong></td>
<td><strong>0.572</strong></td>
</tr>
</tbody>
</table>

4.5. **Pascal3D+**

We perform experiments on Pascal3D+ and show quantitative and qualitative results in Tab. 3, Tab. 4 and Fig. 8.

**Ablation Study.** We perform a similar ablation to Pix3D on Pascal3D+. The results corroborate the findings from Pix3D; we verify the effectiveness of our proposed techniques both quantitatively (Tab. 3) and qualitatively (Fig. 8).

**SOTA Comparison.** We first compare our method with Cat3D and SSMP. Due to the instability of adversarial regularization and the lack of synthetic pretraining, both methods cannot scale up well to more diverse real-world scenarios and compare poorly to our method as demonstrated in Tab. 3. In Tab. 4, we further compare to SS3D which also uses synthetic pretraining. Because SS3D cannot predict viewpoints, we evaluate reconstructed shapes via the brute-force pose alignment for both our method and SS3D. In the quantitative comparison, our method outperforms SS3D by a large margin; we also learn better global structures and more accurate local details in the qualitative examples, as shown in Fig. 8.

4.6. **OpenImages**

We perform experiments on OpenImages and show a qualitative comparison to SS3D in Fig. 6 across various categories. As shown in the figure, our method performs favorably to SS3D by reconstructing more accurate shapes both globally and locally. These results verify the effectiveness and scalability of our method.

4.7. **Limitations**

Although the results we achieve are promising, we find our method still can not work well for categories that are often occluded, or for largely deformable categories. We also do not handle the shape misalignment in our semantic constraint explicitly, which can be detrimental for categories with complex/deformable shapes. We think future exploration along these directions would be exciting.

5. **Conclusion**

We present a novel model that reconstructs 3D object shapes over real-world single-view images in a scalable way. Our model is driven by two key techniques, the CLIP-based semantic constraint and the local geometric constraint. These two techniques significantly benefit the global shape understanding and local geometry reconstruction. They enable us to achieve SOTA performance on three challenging real-world datasets containing various objects.

**Acknowledgement:** This work was supported in part by NIH R01HD104624-01A1 and a gift from Google.
References


[27] Ricardo Martin-Brualla, Noha Radwan, Mehdil SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duck-


[51] Guandao Yang, Yin Cui, Serge Belongie, and Bharath Hariharan. Learning single-view 3d reconstruction with limited


