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

Zixuan Huang[1] Varun Jampani[2] Anh Thai[1] Yuanzhen Li[2]
Stefan Stojanov[1] James M. Rehg[1]
[1]Georgia Institute of Technology, [2]Google Research

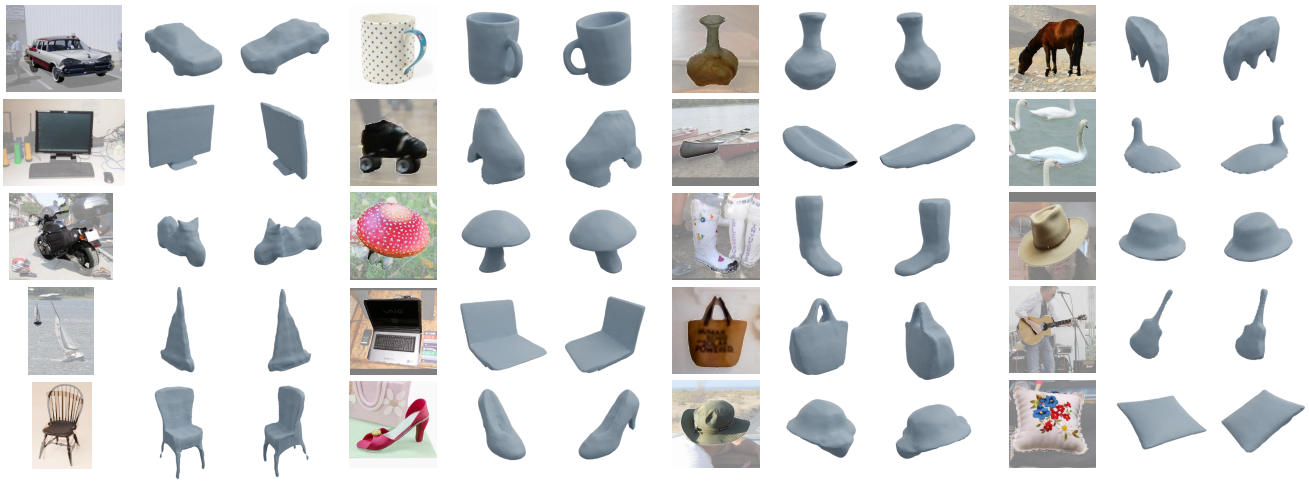

Figure 1. We propose a method that reconstructs 3D object shape from single view real-world images. Our method learns high-quality reconstruction through ***single-view supervision without known viewpoint*** and can reconstruct shapes of various objects.

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.[1]

[1]Project website at: https://zixuanh.com/projects/shapeclipper.html

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 [4, 9, 11, 28, 36, 41, 43–45, 48, 49, 53] or multi-view supervision [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 [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 [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

Figure 2. **CLIP-based semantic neighbors.** Samples that have similar CLIP encodings often have similar shapes. Note the viewpoint variability in the neighbors.

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 shapes (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

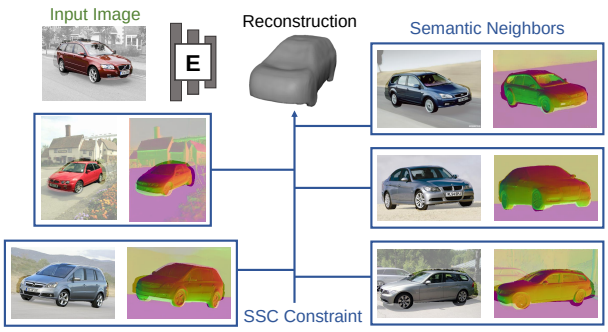

Figure 3. **Semantic-based Shape Consistency (SSC) Constraint.** We find the semantic neighbors of the input image across the training set and use these neighbors to regularize the shape learning.

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 images, 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.

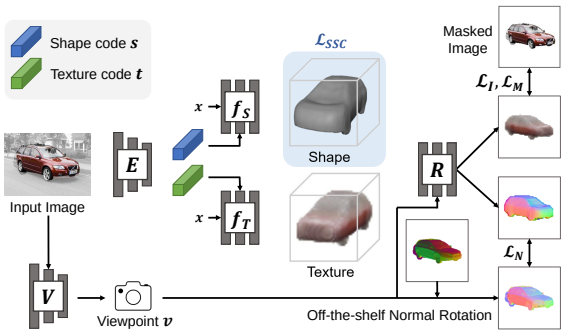

Figure 4. **Network overview.** Given the input image, the encoder $\boldsymbol{E}$ infers the shape latent code $\boldsymbol{s}$ and texture latent code $\boldsymbol{t}$. By conditioning these two codes upon shape MLP $\boldsymbol{f_S}$ and texture MLP $\boldsymbol{f_T}$, we obtain the shape and texture reconstruction of the input object. On the other hand, the viewpoint estimator $\boldsymbol{V}$ estimates the input viewpoint $\boldsymbol{v}$. The differentiable volume renderer $\boldsymbol{R}$ then renders shape and texture fields under the estimated viewpoint, so that we can compute the reconstruction loss $\mathcal{L}_I$ 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.

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 $\boldsymbol{E}$ takes a segmented image $I \in \mathbb{R}^{h\times w\times 3}$ as input and infers the shape latent code $\boldsymbol{s} \in \mathbb{R}^l$ and the texture latent code $\boldsymbol{t} \in \mathbb{R}^l$. 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, $\boldsymbol{f_S} : \mathbb{R}^3 \rightarrow \mathbb{R}$ and $\boldsymbol{f_T} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, which predict signed distance function (SDF) and RGB values of queried 3D coordinates respectively. The MLPs are conditioned on the latent codes, with a similar design to VolSDF [54]. Specifically, the shape MLP $\boldsymbol{f_S}$ is conditioned on $\boldsymbol{s}$ and the texture MLP $\boldsymbol{f_T}$ is conditioned on $\boldsymbol{t}$.

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.

Model	[17]	[18]	[24]	[38]	[57]	[10]	[22]	[14]	[13]	[31]	[46]	[29]	[15]	[55]	[1]	Ours
3D Rep.	M	M	S	M	M	M	M	V	M	P	D	M	S	M	O	S
Viewpoint Free	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Diverse R-Res.	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓

Viewpoint estimator. The viewpoint estimator $\boldsymbol{V}$ 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. $\boldsymbol{v} = [\cos\boldsymbol{\gamma}, \sin\boldsymbol{\gamma}]$ where $\boldsymbol{\gamma}$ denotes the three Euler angles.

Differentiable renderer. We use a volume renderer $\boldsymbol{R}$ 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, $\boldsymbol{R}(\boldsymbol{f_S}, \boldsymbol{f_T}, \boldsymbol{v})$, which maps the implicit functions and the viewpoint into image $\hat{I}$, mask $\hat{M}$ and surface normal $\hat{N}$.

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:

$$\mathcal{L}_I = \|I - \hat{I}\|_2^2, \ \mathcal{L}_M = 1 - IOU(M, \hat{M}), \tag{1}$$

$$IOU(M, \hat{M}) = \frac{\sum_p M^p \cdot \hat{M}^p}{\sum_p M^p + \hat{M}^p - M \cdot \hat{M}^p}. \tag{2}$$

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 $\{\boldsymbol{c}_i\}$. 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 $\{\boldsymbol{c}_i\}$ 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 $\boldsymbol{s}$ and $\boldsymbol{f_S}$. 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 $\boldsymbol{E}$ then predicts the latent texture code $\boldsymbol{t}_k$ for I_k, which is used to generate its texture

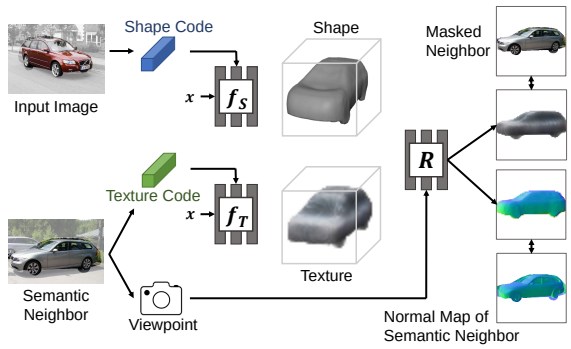

Figure 5. **Semantic-based Shape Consistency (SSC) constraint.** We improve shape learning via the SSC constraint. The shape reconstructed for the input object has to explain its CLIP semantic neighbor as well.

function $\boldsymbol{f_{T_k}}$. We also obtain the viewpoint prediction $\boldsymbol{v}_k$ through the viewpoint estimator $\boldsymbol{V}$. We can then combine the input shape $\boldsymbol{f_S}$ with the neighbor texture $\boldsymbol{f_{T_k}}$ and viewpoint $\boldsymbol{v}_k$, and render them into an image $\hat{I}'$, a mask $\hat{M}'$ and a normal map $\hat{N}'$. Namely, $(\hat{I}', \hat{M}', \hat{N}') = \boldsymbol{R}(\boldsymbol{f_S}, \boldsymbol{f_{T_k}}, \boldsymbol{v}_k)$. By replacing input maps in Eq. (1) with semantic neighbors, we can obtain the SSC losses for this sample in the same form:

$$\mathcal{L}_{SSC_I} = \|I_k - \hat{I}'\|_2^2,\ \mathcal{L}_{SSC_M} = 1 - IOU(M_k, \hat{M}'). \quad (3)$$

3.3. Geometric Constraint

We further propose to facilitate shape learning via geometric constraints to encourage the model to learn better low-level geometric reasoning. The idea here is to estimate the surface normal of our implicit shape, and make it consistent with the surface normal prediction from off-the-shelf models. The off-the-shelf normal estimator we use is Omnidata [7], which is a state-of-the-art normal estimator. Recent work has also proven its effectiveness for multi-view scene reconstruction [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]:

$$\mathcal{L}_N = \beta \cdot \|RN - \hat{N}\|_1 - cos(RN, \hat{N}), \quad (4)$$

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 $\mathcal{L}_N$ 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, $\mathcal{L}_{SSC_N}$. This can be calculated similarly to $\mathcal{L}_N$, 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 [27] before fed into the MLPs. The conditioning of the MLPs is achieved via concatenation, and the shape latent code is additionally skip-connected to the first and the second hidden layers of the shape MLP. Following VolSDF, we condition the texture MLP with the shape MLP's last-layer feature as well. The differentiable renderer we use renders the volumes by uniformly sampling 64 points along each ray.

Loss function. Our overall loss function is a summation of the reconstruction loss and the SSC losses (with our geometric constraint included):

$$\mathcal{L}_{recon} = \mathcal{L}_I + \lambda_1 \mathcal{L}_M + \lambda_2 \mathcal{L}_N, \quad (5)$$

$$\mathcal{L}_{SSC} = \mathcal{L}_{SSC_I} + \lambda_1 \mathcal{L}_{SSC_M} + \lambda_2 \mathcal{L}_{SSC_N}, \quad (6)$$

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{SSC}. \quad (7)$$

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].

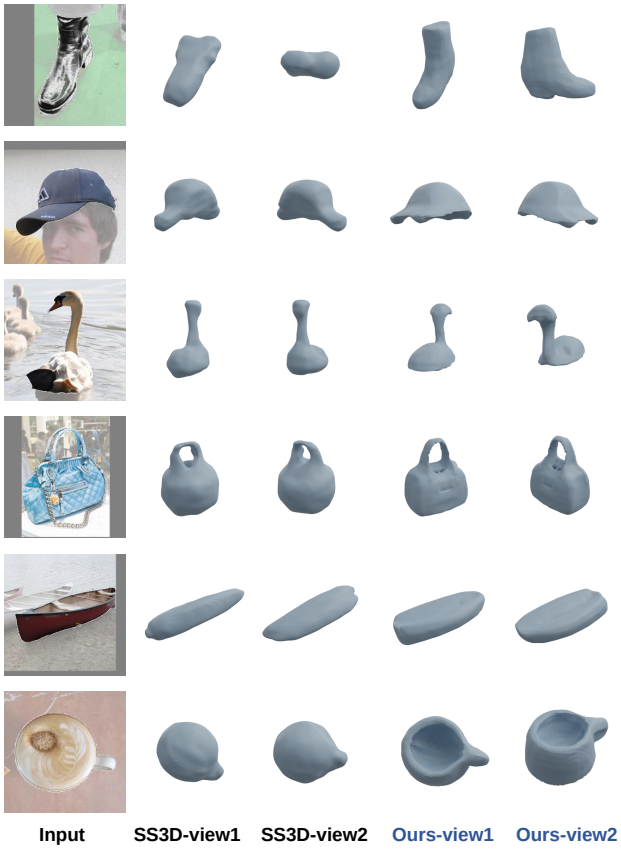

Figure 6. **Qualitative comparison on OpenImages.** Our method learns both better global 3D structure and shape details on various categories.

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:

$$d(S_1, S_2) = \frac{1}{2|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x-y\|_2 + \frac{1}{2|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|x-y\|_2 \tag{8}$$

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.

Methods	FS@1↑	FS@5↑	FS@10↑	CD↓
w/o $\mathcal{L}_{SSC}$	0.0958	0.4309	0.7093	0.749
w/o normal	0.0815	0.3913	0.6982	0.766
w/o noise-tol	0.1277	0.5319	0.7861	0.640
Ours	**0.1317**	**0.5473**	**0.8002**	**0.618**
Cat3D [15]	0.0960	0.4410	0.7262	0.679
SSMP [55]	0.0948	0.4261	0.7168	0.707

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 $\mathcal{L}_{SSC}$' 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.

[2]SSMP has an optional test-time optimization to further convert voxels into meshes. Using the authors' public implementation of this optimization did not improve their results in our experiments, so we use their voxel prediction for evaluation.

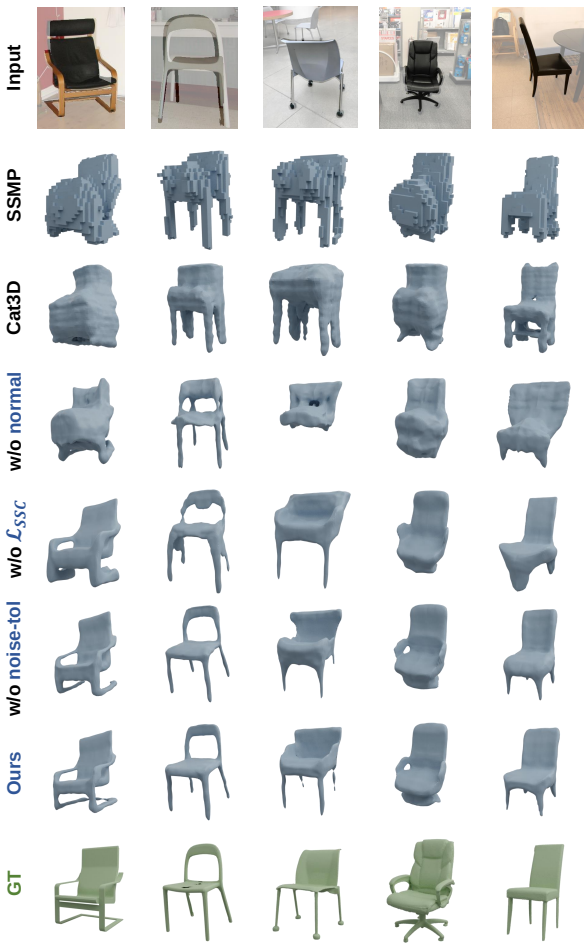

Figure 7. **Qualitative comparison on Pix3D.** Our method learns better global 3D structure and shape details than other baselines.

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 pretraining for any methods.

Table 3. **Quantitative results on Pascal3D+.** Our method performs favorably to baselines and other SOTA methods.

Methods	FS@1↑	FS@5↑	FS@10↑	CD↓
w/o $\mathcal{L}_{SSC}$	0.1363	0.5268	0.7307	0.898
w/o normal	0.1185	0.4648	0.6712	0.952
w/o noise-tol	**0.1548**	0.5875	0.7874	0.707
Ours	0.1519	**0.5914**	**0.7981**	**0.693**
Cat3D [15]	0.0858	0.3977	0.6155	1.118
SSMP [55]	0.1014	0.4366	0.6614	1.000

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

Methods	FS@1↑	FS@5↑	FS@10↑	CD↓
SS3D [1]	0.0533	0.4879	0.7702	0.696
Ours	**0.0585**	**0.5388**	**0.8507**	**0.572**

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.

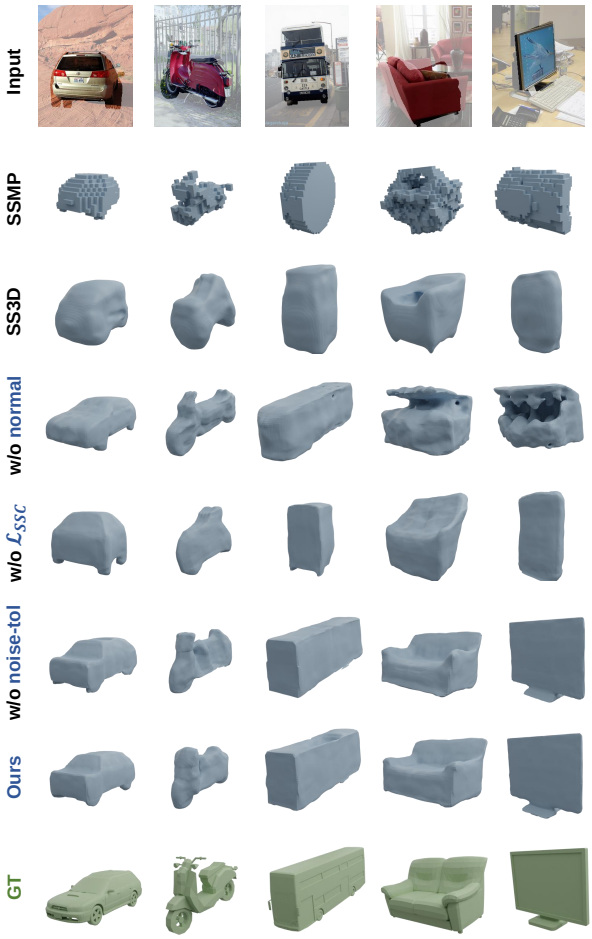

Figure 8. **Qualitative comparison on Pascal3D.** Our method learns better global 3D structure and shape details than other baselines.

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

[1] Kalyan Vasudev Alwala, Abhinav Gupta, and Shubham Tulsiani. Pre-train, self-train, distill: A simple recipe for supersizing 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3773–3782, 2022. 1, 3, 4, 5, 7, 8, 13

[2] Lucas Beyer, Alexander Hermans, and Bastian Leibe. Biternion nets: Continuous head pose regression from discrete training labels. In *German Conference on Pattern Recognition*, pages 157–168. Springer, 2015. 4

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

[4] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 1, 3

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4

[7] Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10786–10796, 2021. 2, 5

[8] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, pages 402–411. IEEE, 2017. 1

[9] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 3

[10] Shubham Goel, Angjoo Kanazawa, and Jitendra Malik. Shape and viewpoint without keypoints. In *European Conference on Computer Vision*, pages 88–104. Springer, 2020. 1, 3, 4

[11] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3, 6

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[13] Paul Henderson and Vittorio Ferrari. Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision*, pages 1–20, 2019. 3, 4

[14] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9984–9993, 2019. 1, 3, 4

[15] Zixuan Huang, Stefan Stojanov, Anh Thai, Varun Jampani, and James M Rehg. Planes vs. chairs: Category-guided 3d shape learning without any 3d cues. In *European Conference on Computer Vision*, pages 727–744. Springer, 2022. 1, 3, 4, 6, 7, 8

[16] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. *arXiv preprint arXiv:1810.09381*, 2018. 1, 3

[17] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 1, 3, 4

[18] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9778–9787, 2019. 1, 3, 4

[19] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018. 1, 3

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 6, 13

[22] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3d reconstruction via semantic consistency. In *European Conference on Computer Vision*, pages 677–693. Springer, 2020. 1, 3, 4

[23] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 3

[24] Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. Sdf-srn: Learning signed distance 3d object reconstruction from static images. *arXiv preprint arXiv:2010.10505*, 2020. 1, 3, 4, 6

[25] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7708–7717, 2019. 1, 3

[26] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 6

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

worth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *arXiv preprint arXiv:2008.02268*, 2020. 5

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

[29] Tom Monnier, Matthew Fisher, Alexei A. Efros, and Mathieu Aubry. Share With Thy Neighbors: Single-View Reconstruction by Cross-Instance Consistency. In *ECCV*, 2022. 1, 3, 4

[30] Siva Karthik Mustikovela, Varun Jampani, Shalini De Mello, Sifei Liu, Umar Iqbal, Carsten Rother, and Jan Kautz. Self-supervised viewpoint learning from image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3971–3981, 2020. 6

[31] KL Navaneet, Ansu Mathew, Shashank Kashyap, Wei-Chih Hung, Varun Jampani, and R Venkatesh Babu. From image collections to point clouds with self-supervised shape and pose networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1132–1140, 2020. 1, 3, 4

[32] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 1, 3

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2

[34] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 2

[35] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[36] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019. 1, 3

[37] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 5

[38] Alessandro Simoni, Stefano Pini, Roberto Vezzani, and Rita Cucchiara. Multi-category mesh reconstruction from image collections. *arXiv preprint arXiv:2110.11256*, 2021. 1, 4

[39] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 6

[40] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019. 6, 12

[41] Anh Thai, Stefan Stojanov, Vijay Upadhya, and James M Rehg. 3d reconstruction of novel object shapes from single images. In *2021 International Conference on 3D Vision (3DV)*, pages 85–95. IEEE, 2021. 1, 3, 6

[42] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2626–2634, 2017. 1, 3

[43] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 1, 3

[44] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1042–1051, 2019. 1, 3

[45] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. *Advances in neural information processing systems*, 30, 2017. 1, 3

[46] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2020. 1, 3, 4

[47] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE, 2014. 6

[48] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2690–2698, 2019. 1, 3

[49] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019. 1, 3

[50] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. *arXiv preprint arXiv:1612.00814*, 2016. 1, 3

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

pose supervision. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 86–101, 2018. 1, 3

[52] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3973–3981, 2015. 4

[53] Yuan Yao, Nico Schertler, Enrique Rosales, Helge Rhodin, Leonid Sigal, and Alla Sheffer. Front2back: Single view 3d shape reconstruction via front to back prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 531–540, 2020. 1, 3

[54] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 3, 4, 12

[55] Yufei Ye, Shubham Tulsiani, and Abhinav Gupta. Shelf-supervised mesh prediction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8843–8852, 2021. 1, 3, 4, 6, 7, 8, 13

[56] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *arXiv preprint arXiv:2206.00665*, 2022. 5, 12, 13

[57] Junzhe Zhang, Daxuan Ren, Zhongang Cai, Chai Kiat Yeo, Bo Dai, and Chen Change Loy. Monocular 3d object reconstruction with gan inversion. In *European Conference on Computer Vision*, pages 673–689. Springer, 2022. 1, 3, 4

A. Generalization performance

Performance on unseen categories. We train on 6 categories (car, chair, diningtable, motorbike, train, tvmonitor) of Pascal3D+ and test on other unseen categories (see Fig. 9 (a)). We find our model can generalize to categories that are highly related to at least one training category (e.g. sofa - average CD 0.571), and does not generalize as well to categories less related to training categories (e.g. bottle - average CD 1.020).

We further quantify the relationship between generalization performance and semantic relevance. We analyze the correlation between reconstruction error and minimum CLIP distance from each test sample to training images. As in Fig. 10 (b), there exists a clear positive correlation (Pearson coefficient $\rho = 0.53$) between the two variables. This verifies our model often generalizes better to samples that are more semantically related to the seen categories.

Figure 9. (a) Reconstruction on unseen categories. (b) Inference of our Pix3D-trained model on CO3D chairs.

Direct inference on in-the-wild data. Our method can also reconstruct faithful shapes under reasonable domain gaps. We test our Pix3D model directly on CO3D chair images (without fine-tuning) and find most reconstructions are reasonable. See Fig. 9 (b) for examples.

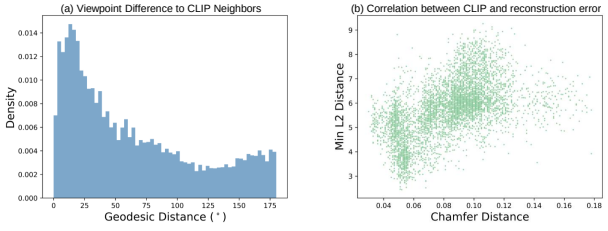

Figure 10. (a) Histogram of viewpoint distance from query images to 5 CLIP neighbors on Pix3D. (b) Correlation between reconstruction error and minimum CLIP distance on unseen categories.

B. Additional Analysis of the Model

Viewpoint robustness of CLIP embeddings. We quantitatively evaluate the viewpoint robustness of CLIP embeddings on Pix3D chairs. In our experiments, CLIP can find a significant number of neighbors with distinct viewpoints (see Fig. 10 (a)). Geodesic distance is the minimal angular difference between two rotations. The average geodesic distance from query images to CLIP neighbors is 64°, and 67% of the query images have at least one neighbor with distinct pose (at least 90°away).

Robustness to corrupted masks. To evaluate the performance under corrupted masks, we replace the Pix3D masks with masks corrupted by perlin noise and then train/test our models under different level of pixel corruption percentages. Under 0/5/10/20/30/50% corruption level, the chamfer distance is 0.612/0.626/0.632/0.651/0.689/0.763 respectively. This experiment shows that our model is not significantly affected by mild to moderate mask inaccuracy.

Performance with GT viewpoints. We further evaluate our model when GT viewpoints are given during training. An image can be explained by infinite combinations of shapes and viewpoints. When GT viewpoints are given, such entanglement is resolved and the learning will be much easier. Our model trained with GT viewpoints obtains average CD of 0.418 on Pix3D (vs. 0.612 w/o GT viewpoint).

Additional discussion on retrieval methods. Comparing reconstruction methods to retrieval methods has been one of the central topics in the area of single-view shape reconstruction [40]. Based on the finding about CLIP's relationship to shape in our paper, it would be natural to consider the retrieval baseline using CLIP. While retrieving shapes with CLIP is an interesting direction, we would like to emphasize that it is not directly comparable to our proposed reconstruction method. Retrieval methods require a **large paired** image-3D shape database, similar to the non-scalable fully supervised 3D reconstruction setting. In contrast, our method only requires single-view 2D images during training, allowing it to learn to reconstruct objects from datasets like OpenImages for which there are no paired 3D shapes. Such datasets without any geometric annotations are our main application domain, and retrieval methods cannot be applied to these datasets.

C. Additional Implementation Details.

Implicit representation and rendering. The surface representation and texture rendering follow [54, 56]. We use an implicit SDF field and convert it to densities for volumetric rendering. The conversion from SDF to densities is done via the Cumulative Distribution Function (CDF) of the Laplace distribution:

$$\sigma(s) = \begin{cases} \frac{1}{\beta} \cdot \frac{1}{2} \exp(\frac{s}{\beta}) & \text{if } s \leq 0 \\ \frac{1}{\beta} \cdot (1 - \frac{1}{2} \exp(-\frac{s}{\beta})) & \text{if } s > 0 \end{cases}, \tag{9}$$

where s is the SDF. Because our focus is shape, and learning radiance is challenging without viewpoint annotation, we represent texture as an RGB field without any view-dependency. The surface normal rendering follows

MonoSDF [56], where the local normal vectors are estimated by the gradient of the SDF field and aggregated via the standard volume rendering.

Uniform viewpoint prior. We use a uniform prior to regularize the viewpoint learning, which helps to prevent the rotation estimation from degeneration. Specifically, for each minibatch during training, we estimate the empirical distribution of the predicted azimuth. We then minimize the Earth-Mover Distance (EMD) between the empirical azimuth distribution and a uniform prior within $[0°, 360°]$.

D. Additional Results on OpenImages

We show more reconstruction results of our model trained on all 53 OpenImages [21] categories used in [55]. We further compare with SS3D [1] qualitatively, where our model demonstrates state-of-the-art reconstruction performance. Note the training is category-specific and the performance may be further improved by training a joint model via distillation [1], which is parallel to our research direction here.

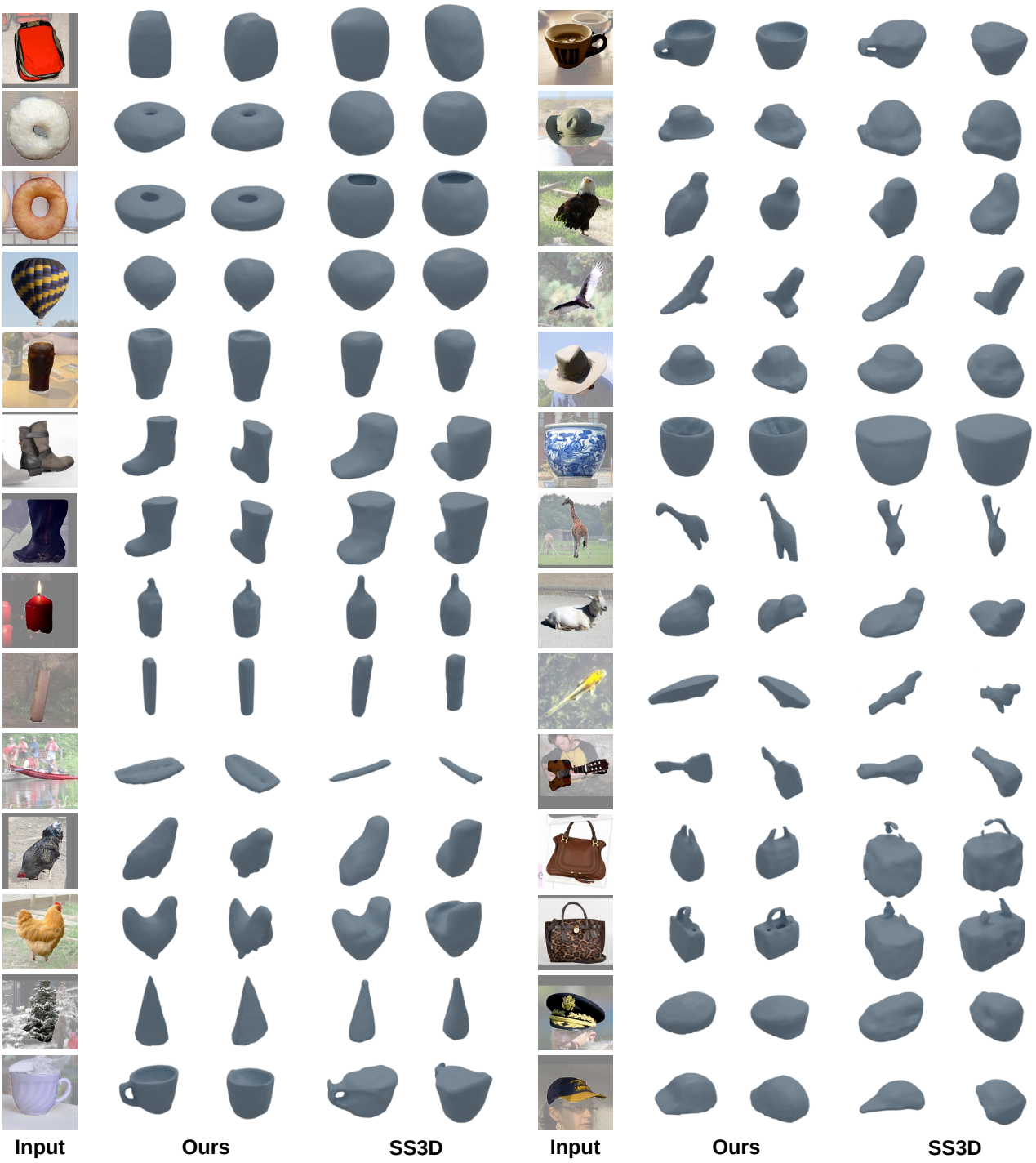

Figure 11. Additional qualitative results and comparison on full OpenImages.

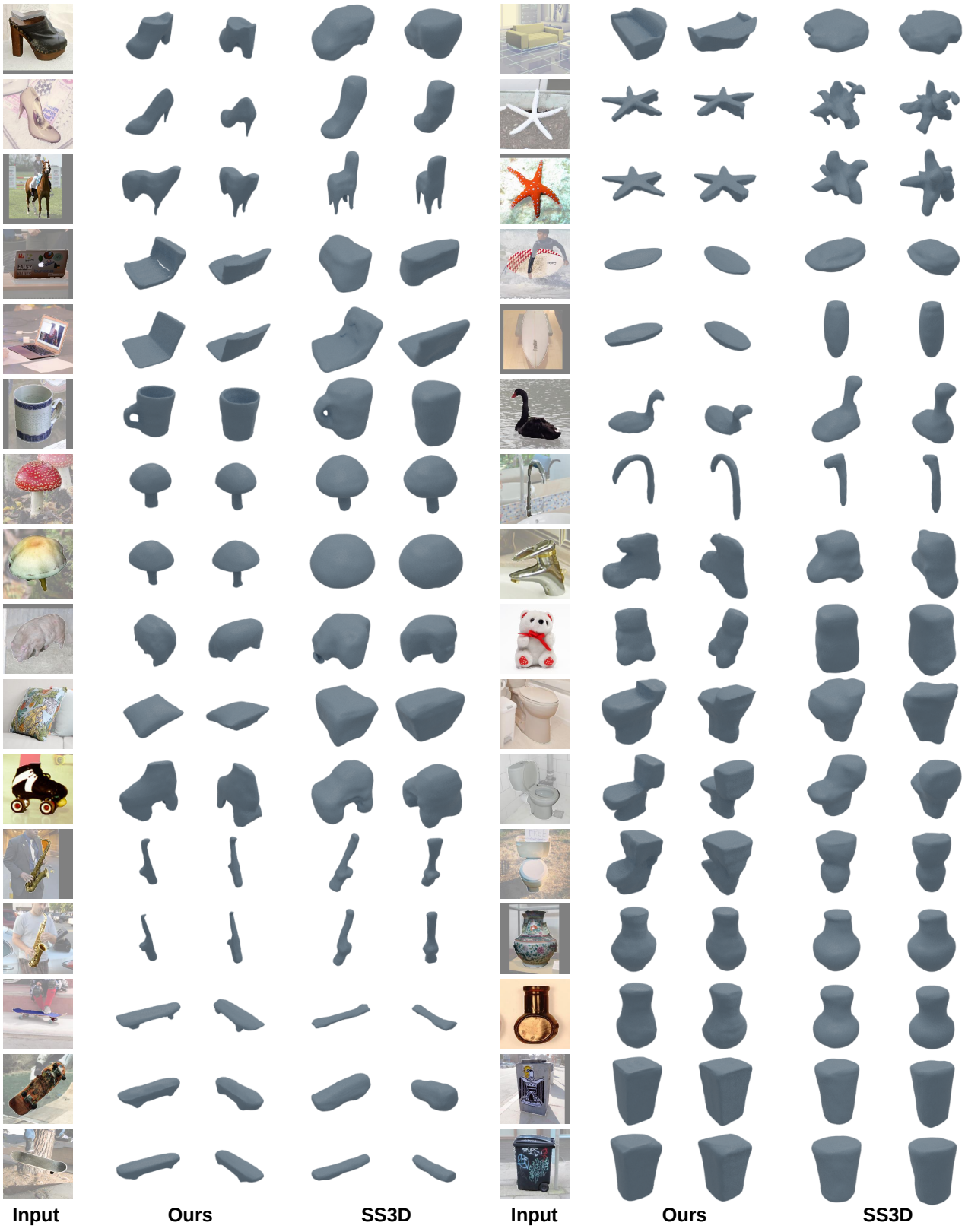

Figure 12. Additional qualitative results and comparison on full OpenImages.