Self-supervised Single-view 3D Reconstruction via Semantic Consistency

Xueting Li[⋆1], Sifei Liu[2], Kihwan Kim[2], Shalini De Mello[2], Varun Jampani[2], Ming-Hsuan Yang[1], and Jan Kautz[2]

[1] University of California, Merced
[2] NVIDIA

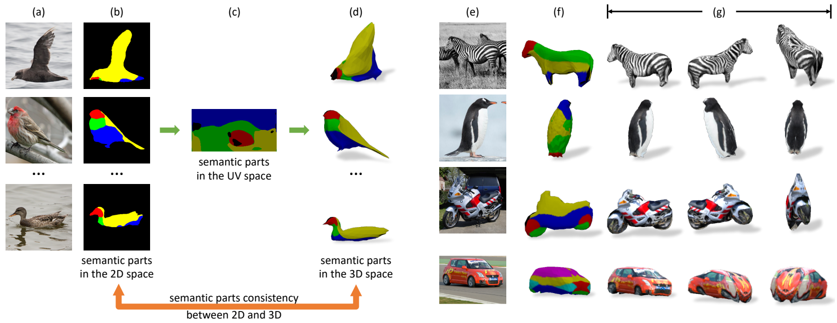

(a)
(b)
(c)
(d)
(e)
(f)
(g)
...
...
semantic parts in the UV space
...
semantic parts in the 2D space
semantic parts in the 3D space
semantic parts consistency between 2D and 3D

Fig. 1: **Self-supervision with semantic part consistency (a–d):** (a) Images of different objects in the same category (e.g., birds in this example). (b) Semantic part segmentation for each image learned via self-supervision. (c) Canonical semantic UV map for the category. (d) Semantic part segmentation on meshes. **Single-view 3D Mesh reconstruction (e–g):** Reconstruction (inference) of each single-view image (e) is demonstrated in (g), along with semantic labels of the mesh in (f).

Abstract. We learn a self-supervised, single-view 3D reconstruction model that predicts the 3D mesh shape, texture and camera pose of a target object with a collection of 2D images and silhouettes. The proposed method does not necessitate 3D supervision, manually annotated keypoints, multi-view images of an object or a prior 3D template. The key insight of our work is that objects can be represented as a collection of deformable parts, and each part is semantically coherent across different instances of the same category (e.g., wings on birds and wheels on cars). Therefore, by leveraging part segmentation of a large collection of category-specific images learned via self-supervision, we can effectively enforce semantic consistency between the reconstructed meshes and the original images. This significantly reduces ambiguities during joint prediction of shape and camera pose of an object, along with texture. To the best of our knowledge, we are the first to try and solve the single-view reconstruction problem without a category-specific template mesh or semantic keypoints. Thus our model can easily generalize to various object categories without such labels, e.g., horses, penguins, etc. Through a

[⋆] Work done during an internship at NVIDIA.

variety of experiments on several categories of deformable and rigid objects, we demonstrate that our unsupervised method performs comparably if not better than existing category-specific reconstruction methods learned with supervision. More details can be found at the project page https://sites.google.com/nvidia.com/unsup-mesh-2020.

Keywords: 3D from Single Images; Unsupervised Learning

1 Introduction

Recovering both 3D shape and texture, and camera pose from 2D images is a highly ill-posed problem due to its inherent ambiguity. Existing methods resolve this task by utilizing various forms of supervision such as ground truth 3D shapes [3, 34, 33], 2D semantic keypoints [15], shading [11], category-level 3D templates [18] or multiple views of each object instance [40, 17, 35, 27]. These types of supervision signals require tedious human effort, and hence make it challenging to generalize to many object categories that lack such annotations. On the other hand, learning to reconstruct by not using any 3D shapes, templates, or keypoint annotations, i.e., with only a collection of single-view images and silhouettes of object instances, remains challenging. This is because the reconstruction model learned without the aforementioned supervisory signals leads to erroneous 3D reconstructions. A typical failure case is caused by the "camera-shape ambiguity", wherein, incorrectly predicted camera pose and shape result in a rendering and object boundary that closely match the input 2D image and its silhouette, as shown in Figure 2 (c) and (d).

Interestingly, humans, even infants who have never been taught about objects in a category, tend to mentally reconstruct objects in that category by perceiving them as a combination of several basic parts, e.g., a bird has two legs, two wings, and one head, etc., and use the parts to associate all the divergent instances of the category. By observing object parts, humans can also roughly infer the relative camera pose and 3D shape of any specific instance. In computer vision, a similar intuition is formulated by the deformable parts model, where objects are represented as a set of parts arranged in a deformable configuration [7, 24].

Inspired by this intuition, we learn a single-view reconstruction model from a collection of images and silhouettes. We utilize the semantic parts in both the 2D and 3D space, along with their consistency to correctly estimate shape and camera pose. Specifically, we first leverage self-supervised co-part segmentation (SCOPS [14]) to decompose 2D images into a collection of semantic parts (Figure 1(b)). By exploiting the property of *semantic part invariance*, which states that the semantic part label of a point on the mesh surface does not change even when the mesh shape is deformed, we associate the semantic parts of *different* object instances with each other and build a category-level canonical semantic UV map (Figure 1(c)). The semantic part label of each point on the reconstructed mesh surface (Figure 1(d)) is then defined by this canonical semantic UV map. Finally, we resolve the aforementioned "camera-shape ambiguity" and learn the self-supervised reconstruction model by encouraging the consistency of semantic part labels in both the 2D and 3D space (Figure 1, orange arrow). Furthermore, we train our model by iteratively learning (a) instance-level reconstruction and (b) a category-level template mesh from scratch. Thus, our model also does not require a pre-defined 3D template mesh or any other shape prior. Our main contribution is a 3D reconstruction model that is able to:

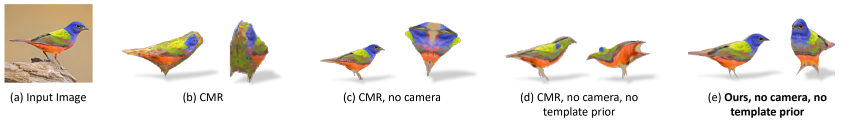
- Conduct single-view mesh reconstruction *without* any of the following forms of supervision: category-level 3D template prior, annotated keypoints, camera pose or multi-view images. In other words, the model can be generalized to other categories which do not have well-defined keypoints, e.g., penguin.
- Leverage the *semantic part invariance* property of object instances of a category as a deformable parts model.
- Learn a category-level 3D shape template from scratch via iterative learning.
- Perform comparably to the state-of-the-art supervised methods [15, 18] trained with either pre-defined templates or annotated keypoints, while also improving the self-supervised semantic co-part segmentation model (SCOPS [14]).

2 Related Work

3D Shape Representation Various representations have been explored for 3D processing tasks, including point clouds [6], implicit surfaces [22, 21], triangular meshes [15, 17, 20, 16, 33, 23, 34] and voxel grids [3, 8, 9, 31, 35, 40, 45, 10]. Among these, while both voxels and point clouds are more friendly to deep learning architectures (e.g., VON [36, 44], PointNet [25, 26], etc), they suffer either from issues of memory inefficiency or are not amenable to differentiable rendering. Hence, in this work, we adopt triangular meshes [15, 17, 20, 16, 33, 23, 34] for 3D reconstruction.

Single-view 3D Reconstruction Single-view 3D reconstruction [3, 8, 9, 31, 35, 40, 45, 6, 11] aims to reconstruct a 3D shape given a single input image. One line of works have explored this ill-posed task with varying degree of supervision. Several methods [33, 23, 34] utilize image and ground truth 3D mesh pairs as supervision. This either requires significant manual annotation effort [38] or is restricted to synthetic data [1]. More recently, a few works [17, 20, 16, 2] avoid 3D supervision by taking advantage of differentiable renderers [17, 20, 2] and the "analysis-by-synthesis" approach, with either multiple views, or known ground truth camera poses.

To further relax constraints on supervision, Kanazawa et al. [15] explored 3D reconstruction from a collection of images of different instances. However, their method still requires annotated 2D keypoints to infer camera pose correctly. It is also the first work to propose a learnable category-level 3D template shape, which, however, needs to be initialized from a keypoint-dependent 3D convex hull. Similar problem settings have also been explored in other methods [29, 37, 12], but with object categories restricted to rigid or structured objects, such as cars or faces. Different from all these works, we target both rigid and non-rigid



(a) Input Image (b) CMR (c) CMR, no camera (d) CMR, no camera, no template prior (e) **Ours, no camera, no template prior**

Fig. 2: **Comparison with baselines.** Each reconstructed mesh is rendered in the original view of the input image and the frontal view of the bird. (b) Shows the result from CMR with camera pose and template prior supervision. (c) Shows CMR with only template prior. (d) Shows CMR without both types of supervision where the model completely fails to learn the texture and shape. In contrast, our model in (e) reconstructs correctly even without supervision from camera pose or a template prior.

objects (e.g., birds, horses, penguins, motorbikes and cars shown in Figure 1 (e)-(g)) and propose a method that jointly estimates a 3D mesh, texture, and camera pose from a single-view image, using only a collection of images with silhouettes as supervisions. In other words, we do not require 3D template priors, annotated keypoints, or multi-view images.

Self-supervised Correspondence Learning Our work is also related to self-supervised cross-instance correspondence learning, via landmarks [30, 43, 13, 28], part segments [4, 14], or canonical surface mapping [18]. We utilize self-supervised co-parts segmentation [14] to enforce semantic consistency, which was originally proposed purely for 2D images. The work of [18] learns a mapping function that maps pixels in 2D images to a predefined category-level template in a self-supervised manner. However, it dose not use the learned correspondence for 3D reconstruction. We show that our work, despite having a focus on 3D reconstruction, outperforms [18] at learning 2D to 3D correspondences as well.

3 Approach

To fully reconstruct the 3D mesh of an object instance from an image, a network should be able to jointly predict the shape and texture of the object, and the camera pose of the image. We start with the existing network from [15] (CMR) as the baseline reconstruction network. Given an input image, CMR extracts the image features using an encoder E and jointly predicts the mesh shape, camera pose and mesh texture by three decoders D_{shape}, D_{camera} and D_{texture}. The mesh shape V is reconstructed by predicting vertex offsets ΔV to a category-specific shape template $\bar{V}$, while the camera pose θ is represented by a weak perspective transformation. To reconstruct mesh textures, the texture decoder outputs a UV texture flow (I_{flow}) that maps pixels from the input image to the UV space. A pre-defined mapping function Φ further maps each pixel in the UV space to a point on the mesh surface.

One of the key elements for the CMR method to perform well is to exploit *mannually annotated semantic keypoints* for (i) precisely pre-computing the ground truth camera pose for each instance, and (ii) estimating a category-level 3D template prior. However, annotating keypoints is tedious, not well-defined

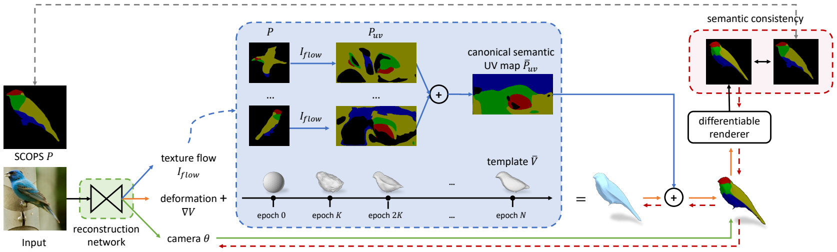


Fig. 3: **Overview.**(a) Green box: The reconstruction network. (b) Red box: Semantic part consistency constraint, see Section 3.1 for more details. (c) Blue box: Computing the canonical semantic UV map and the template shape using the reconstruction network, see Section 3.2. The red dashed arrows show that the gradients from the semantic part consistency constraint facilitate shape and viewpoint estimation.

for most object categories in the world and impossible to generalize to new categories. Thus, we propose a method within a more scalable, but challenging self-supervised setting *without* using manually annotated keypoints to estimate camera pose or a template prior.

Not surprisingly, simply taking out the keypoints supervision, as well as all the related information (i.e., the camera pose and the template prior) from the CMR network makes it unable to predict camera pose and shape correctly, as shown in Figure 2(c) and (d). This is due to the inherent ambiguity of hallucinating 3D meshes from only single-view 2D observations, where the model trivially picks a combination of camera pose and shape that yields the rendering that matches the given image and silhouette. Consider an extreme case, where the model predicts the front view for all instances, but is still able to match the image and silhouette observations by deforming each instance mesh accordingly.

In this work, we propose a framework (Figure 3) designed for self-supervised mesh reconstruction learning, i.e., with only a collection of images and silhouettes as supervision. The framework consists of: (i) A reconstruction network (green box) that has the same architecture as [15] – it consists of an image encoder E and three decoders D_{shape}, D_{camera} and D_{texture} that jointly predict the mesh deformation ΔV, texture flow I_{flow} and camera pose θ for the instance in the image. (ii) A semantic consistency constraint (red box in Figure 3) that regularizes the learning of module (i) and largely resolves the aforementioned "camera-shape ambiguity" under the self-supervised setting. We introduce this module in Section 3.1. (iii) A module that learns the canonical semantic UV map and category-level template from scratch (blue box in Figure 3). This module is iteratively trained with module (i) and discussed in Section 3.2.

3.1 Resolving Camera-Shape Ambiguity via Semantic Consistency

In this section, we show the key to solving the "camera-shape ambiguity" is to make use of the semantic parts of object instances in both 3D and 2D. Specifically, we exploit the fact that (i) in the 2D space, self-supervised co-part segmentation [14] provides correct part segments for a majority of the object instances, even for those with large shape variations (see Figure 1(b)); and (ii) in the 3D space, semantic parts are invariant to mesh deformations, i.e., the semantic part label of a specific point on the mesh surface is consistent across all reconstructed instances of a category. We demonstrate that this *semantic part invariance* allows us to build a category-level semantic UV map, namely the canonical semantic UV map, shared by all instances, which in turn allows us to assign semantic part labels to each point on the mesh. By enforcing consistency between the canonical semantic map and an instance's part segmentation in the 2D space, the camera-shape confusion can be largely resolved.

Part Segmentation in 2D via SCOPS [14] SCOPS is a self-supervised method that learns semantic part segmentation from a collection of images of an object category (see Figure 1(b)). The model leverages concentration and equivariance loss functions, as well as part basis discovery to output a probabilistic map w.r.t. the discovered parts that are semantically consistent across different object instances. We discuss in the supplementary as to how, besides generalizing SCOPS to reconstructing objects, our model also improves SCOPS in return.

Part Segmentation in 3D via Canonical Semantic UV Map Given the semantic part segmentation of 2D images estimated by SCOPS, how can we obtain the semantic part labels for each point on the mesh surface? One intuitive way is to obtain a mapping from the 2D image space to the 3D shape space. Therefore, we propose to first utilize the learned texture flow I_{flow} by our reconstruction network that naturally forms a mapping from the 2D image space to the UV texture space, and then further map the semantic labels from the UV space to the mesh surface by the pre-defined mapping function $\varPhi$. We denote the semantic part segmentation of image i as $P^i \in \mathbb{R}^{H \times W \times N_p}$ (see Figure 3 in the blue bbox), where H and W are the height and width of the image, respectively and N_p is the number of semantic parts. By mapping P^i from the 2D image space to the UV space using the learned texture flow, we obtain a "semantic UV map" denoted as $P^i_{\text{uv}} \in \mathbb{R}^{H_{\text{uv}} \times W_{\text{uv}} \times N_p}$, where H_{uv} and W_{uv} are the UV map's height and width, respectively.

Ideally, all instances should result in the same semantic UV map – the canonical semantic UV map for a category, regardless of shape differences of instances. This is because: (i) the *semantic part invariance* states that the semantic part labels assigned to each point on the mesh surface are consistent across different instances; and (ii) the mapping function $\varPhi$ that maps pixels from the UV space to the mesh surface is pre-defined and independent of deformations in the 3D space, such as face location or area changes. Thus, the semantic part labels of pixels in the UV map should also be consistent across different instances.

However, if we directly sample the individual P^i via the learned texture flow I_{flow}, the obtained semantic UV maps are indeed very different between instances, as shown in Figure 3 (blue box). This is caused by the fact that (i) the part segmentation predictions produced by the self-supervised SCOPS method

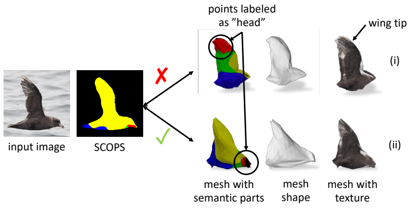


Fig. 4: **Semantic part invariance.** (i) Incorrect reconstruction without semantic part consistency. (ii) Reconstruction with consistency.

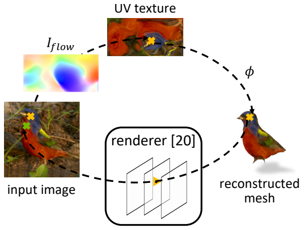


Fig. 5: **The process of texture cycle consistency constraint computation.**

are noisy, and (ii) texture flow prediction is highly uncertain for the invisible faces of the reconstructed mesh. Therefore, we approximate the canonical semantic UV map, denoted as $\bar{P}_{\text{uv}}$ by aggregating the individual semantic UV maps:

$$\bar{P}_{\text{uv}} = \frac{1}{|\mathcal{U}|} \sum_{i \in \mathcal{U}} I^i_{\text{flow}}(P^i), \tag{1}$$

where $I^i_{\text{flow}}(P^i)$ indicates the sampling of P^i by I_{flow} and $\mathcal{U}$ is a subset of selected training samples with accurate texture flow prediction (details of the selection process can be found in the appendix). Through this aggregation process, $\bar{P}_{\text{uv}}$ produces a mean semantic UV map, which effectively eliminates outliers (i.e., instances with incorrect SCOPS), as well as the noisy pixel-level predictions.

Semantic Consistency between 2D and 3D As mentioned above, because our model learns via self-supervision and only relies on images and silhouettes that do not provide any semantic part information, it suffers from the "camera-shape ambiguity" introduced in Section 1. Take row (i) in Figure 4 as an example. The model erroneously forms the wing's tip in the reconstructed bird by deforming the mesh faces assigned to the "head part" (colored in red). This incorrect shape reconstruction, associated with an incorrect camera pose, however, can yield a rendering that matches the observed image and its silhouette.

This ambiguity, although is not easy to spot by only comparing the rendering of the reconstruction with the input image, however, can be identified once the semantic part label for each point on the mesh surface is available. One can tell that the reconstruction in row (i) of Figure 4 is wrong by comparing the rendering of the semantic part labels on the mesh surface and the 2D SCOPS part segmentation. Only when the camera pose and shape are both correct, will the rendering and the SCOPS segmentation be consistent, as shown in row (ii) in Figure 4. This observation inspires us to propose a probability and a vertex-based constraint that facilitate correct camera pose and shape learning by encouraging the consistency of semantic part labels in both 2D images and in the mesh surface.

Probability-based constraint. For each reconstructed mesh instance i, we map the canonical semantic UV map $\bar{P}_{\text{uv}}$ onto its surface by the UV mapping

$\varPhi$ and render it using the predicted camera pose θ^i. We denote the projection from 3D to 2D as $\mathcal{R}$. We constrain the projected probability map to be close to the SCOPS part segmentation probability map P^i by computing the loss:

$$L_{\text{sp}} = \left\| P^i - \mathcal{R}(\varPhi(\bar{P}_{\text{uv}}); \theta^i) \right\|^2 . \tag{2}$$

We empirically found the mean squared error (MSE) metric to be more robust than the Kullback–Leibler divergence for comparing two probability maps.

Vertex-based constraint. We also propose a vertex-based constraint to enhance semantic part consistency (see Figure 2 in the supplementary) by enforcing that 3D vertices assigned a part label p, after being projected to the 2D domain with the predicted camera pose θ^i, align with the area assigned to that part in the input image:

$$L_{\text{sv}} = \sum_{p=1}^{N_p} \frac{1}{|\bar{V}_p|} \text{Chamfer}(\mathcal{R}(\bar{V}_p; \theta^i), Y_p^i), \tag{3}$$

where $\bar{V}_p$ is the set of vertices on a learned category-level 3D *template* $\bar{V}$ (see Section 3.2) with the part label p, Y_p^i is the set of 2D pixels sampled from the part p in the original input image and N_p is the number of parts. Here we use the *Chamfer distance*, because the projected vertices and pixels with the same part label p in the input image do not have a strictly one-to-one correspondence.

Note that, $\bar{V}_p$ is a set of vertices on the category-level shape template $\bar{V}$ as opposed to each instance reconstruction V^i, since using V^i results in a degenerate solution where the network only alters 3D shape to satisfy this vertex-based constraint, rather than the camera pose. Instead, using $\bar{V}$ drives the network towards learning the correct camera pose, in addition to shape.

3.2 Progressive Training

We train the framework in Figure 3 via progressive training based on two considerations: (a) building the canonical semantic UV map, introduced in Section 3.1, requires reliable texture flows to map the SCOPS from images to the UV space. Thus the canonical semantic UV map can only be obtained after the reconstruction network is able to predict texture flow reasonably well, and (b) a canonical 3D shape template [15, 18] is desirable, since it speeds up the convergence of the network [15] and also avoids degenerate solutions when applying the *vertex-based constrain* as introduced in Section 3.1. However, jointly learning the category-level 3D shape template and the instance-level reconstruction network leads to undesired trivial solutions. Thus, we propose an expectation-maximization (EM) style progressive training procedure below. In the E-step, we train the reconstruction network with the current template and canonical semantic UV map fixed, and in the M-step, we update the template and the canonical semantic UV map using the reconstruction network learned in the E-step.

E-step: Learning Instance-specific Reconstruction In the E-step, we fix the canonical semantic UV map as well as the category-level template and train the reconstruction network mainly with the following objectives. (i) A negative IoU objective [16] between the rendered and the ground truth silhouettes for shape learning. (ii) A perceptual distance objective [42, 15] between the rendered and the input RGB images for texture learning. (iii) The probability and vertex-based constraints introduced in Section 3.1 to resolve the "camera-shape ambiguity" under the self-supervised setting. (iv) A texture consistency constraint to facilitate accurate texture flow learning that will be introduced in Section 3.3. Other constraints are included in the appendix. Note that in the first E-step, the template is a sphere and hence the probability and vertex-based constraints are not used.

M-step: Canonical UV Map and Template Learning In the M-step, we compute the canonical semantic UV map introduced in Section 3.1 and learn a category-level template from scratch, i.e., from a sphere primitive. As far as we know, we are the first method that learns a category-level template from scratch. This is in contrast to existing methods [18], where the template is either a readily available instance mesh from the category or is estimated from annotated keypoints [15]. Jointly learning the shape template along with the reconstruction network does not guarantee a meaningful "mean shape" which encapsulates the most representative characteristics of objects in a category. Instead, we propose a feed-forward template learning approach: the template starts out as a sphere and is updated every K training epochs by:

$$\bar{V}_t = \bar{V}_{t-1} + D_{\text{shape}}(\frac{1}{|\mathcal{Q}|}\sum_{i \in \mathcal{Q}} E(I^i)), \tag{4}$$

where $\bar{V}_t$ and $\bar{V}_{t-1}$ are the updated and current templates, respectively, I^i is the input image passed to the image encoder E and D_{shape} is the shape decoder (see the beginning of Section 3). $\mathcal{Q}$ is a set of selected training images with consistent mesh predictions and their selection procedure is discussed in the appendix. The template $\bar{V}_t$ is the mean shape of instances in a category for the current epoch, which enforces a meaningful shape (e.g., the template looks like a bird) rather than an arbitrary form for the category.

3.3 Texture Cycle Consistency Constraint

One issue with the learned texture flow is that the texture of 3D mesh faces with a similar color (e.g., black) can be incorrectly sampled from a single pixel location of the image (See Figure 3 in the supplementary). Thus we introduce a texture cycle consistency objective to regularize the predicted texture flow (i.e., 2D→3D) to be consistent with the camera projection (i.e., 3D→2D). As shown in Figure 5, considering the pixel marked with a yellow cross in the input image, it can be mapped to the mesh surface through the predicted texture flow I_{flow} along with the pre-defined mapping function Φ introduced in Section 3.

Meanwhile, its mapping on the mesh surface can be re-projected back to the 2D image by the predicted camera pose, as shown by the green cross in Figure 5. If the predicted texture flow conforms to the predicted camera pose, the yellow and green crosses would overlap, forming a $2D \rightarrow 3D \rightarrow 2D$ cycle.

Formally, given a triangle face j, we denote the set of input image pixels mapped to this face by texture flow as Ω^j_{in}. We further infer the set of pixels (denoted as Ω^j_{out}) projected from the triangle face j in the rendering operation by taking advantage of the probability map $\mathcal{W} \in \mathcal{R}^{|F| \times (H \times W)}$ in the differentiable renderer [20] where $|F|, H, W$ are the number of faces, height and width of the input image, respectively. Each entry in $\mathcal{W}^m_j$ indicates the probability of face j being projected onto the pixel m. We compute the geometric center of both sets (Ω^j_{in} and Ω^j_{out}), denoted by $\mathcal{C}^j_{\text{in}}$ and $\mathcal{C}^j_{\text{out}}$, respectively as:

$$\mathcal{C}^j_{\text{in}} = \frac{1}{N_c} \sum_{m=1}^{N_c} \Phi(I_{\text{flow}}(\mathcal{G}^m))_j; \quad \mathcal{C}^j_{\text{out}} = \frac{\sum_{m=1}^{H \times W} \mathcal{W}^m_j \times \mathcal{G}^m}{\sum_{m=1}^{H \times W} \mathcal{W}^m_j}, \tag{5}$$

where $\mathcal{G} \in \mathbb{R}^{(H \times W) \times 2}$ is a standard coordinate grid of the projected image (containing pixel location (u, v) values), and Φ is the fixed UV mapping that, along with the texture flow I_{flow} maps pixels from the 2D input image to a mesh face j, as discussed in the beginning of Section 3. N_c is the number of pixels in the input image mapped to each triangular face and × indicates multiplication between two scalars. We constrain the predicted texture flow to be consistent with the rendering operation by encouraging $\mathcal{C}^j_{\text{in}}$ to be close to $\mathcal{C}^j_{\text{out}}$:

$$L_{\text{tcyc}} = \frac{1}{|F|} \sum_{j=1}^{|F|} \left\| \mathcal{C}^j_{\text{in}} - \mathcal{C}^j_{\text{out}} \right\|^2_F. \tag{6}$$

We note that while not targeting 3D mesh reconstruction directly, a similar intuition, but with a different formulation was also introduced in [18].

4 Experimental Results

We first introduce our experimental settings in Section 4.1, and present qualitative evaluations for the bird, horse, motorbike and car categories in Section 4.2. Quantitative evaluations and ablation studies for the contribution of each proposed module are discussed in Section 4.3 and Section 4.4, respectively.

4.1 Experimental Settings

We validate our method on both rigid objects, i.e., *car* and *motorcycle* images from the PASCAL3D+ dataset [39], and non-rigid objects, i.e., *bird* images from the CUB-200-2011 dataset [32], *horse*, *zebra*, *cow* images from the ImageNet dataset [5] and *penguin* images from the OpenImages dataset [19]. We use progressive training (Section 3.2) to learn the model parameters. In each E-step, the reconstruction network is trained for 200 epochs and then used to update the

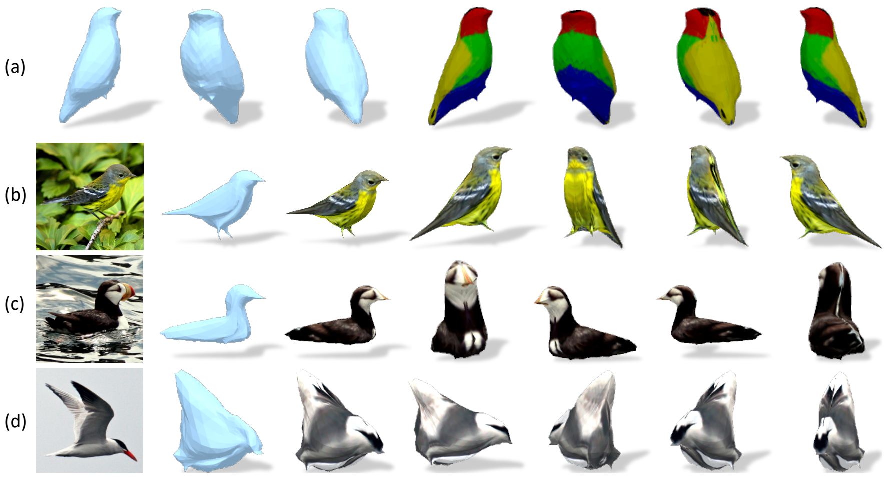


Fig. 6: **Learned template and instance reconstructions from single-view images**. (a) The learned template shape (first three columns) and semantic parts (last four columns). (b)-(d) 3D reconstruction from a single-view image. In each row from left to right, we show the input image, reconstruction rendered using the predicted camera view and from four other views. Please see the results for additional views in the appendix video.

template and the canonical semantic UV map in the M-step. The only exception is in the first round (a round consists of one E and M-step), where we train the reconstruction network without the semantic consistency constraint. This is because, at the beginning of training, I_{flow} is less reliable, which in turn makes the canonical semantic UV map less accurate.

4.2 Qualitative Results

Thanks to the self-supervised setting, our model is able to learn from a collection of images and silhouettes (e.g., horse and cow images [5] and penguin images [19]), which cannot be achieved by existing methods [15, 33, 41, 17] that require extra supervisory signals.

Template and Semantic Parts on 3D Meshes We show the learned templates for the bird, horse, motorbike and car categories in Figure 6 and Figure 7, which capture the shape characteristics of each category, including the details such as the beak and feet of a bird, etc. We also visualize the canonical semantic UV map by showing the semantic part labels assigned to each point on the template's surface. For instance, bird meshes have four semantic parts – head (red), neck (green), belly (blue) and back (yellow) in Figure 6, which are consistent with the part segmentation predicted by SCOPS [14].

Instance 3D Reconstruction We show the results of 3D reconstruction from each single-view image in Figure 6 (b)-(d) and Figure 7 (b). Our model can

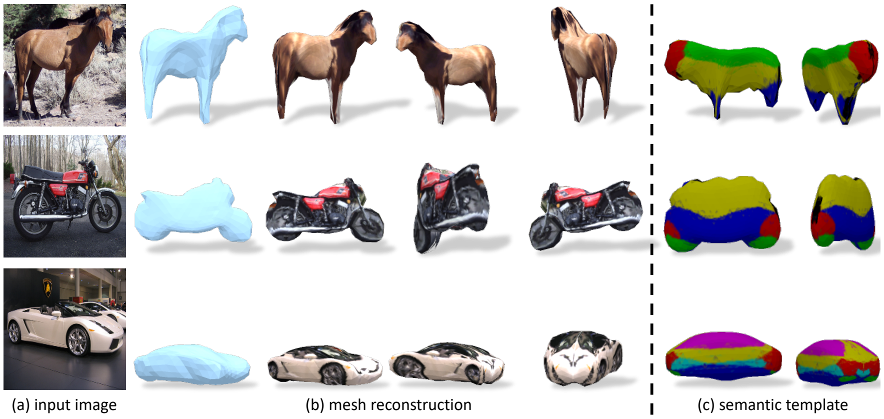


Fig. 7: **More reconstruction results.** Visualization of instance-level reconstructions and semantic templates for the *horse*, *motorbike* and *car* categories.

reconstruct instances from an object category with highly divergent shapes, e.g., a thin bird in (b), a duck in (c) and a flying bird in (d). Our model also correctly maps the texture from each input image onto its 3D mesh, e.g., the eyes of each bird as well as fine textures on the back of the bird. Furthermore, the renderings of the reconstructed meshes under the predicted camera poses (2nd and 3rd columns in Figure 6 and Figure 7) match well with the input images in the first column, indicating that our model accurately predicts the original camera view.

4.3 Quantitative Evaluations

As a self-supervised approach, our model is more practically suited to reconstruct many non-rigid objects, e.g., animals captured in the wild that do not have 3D ground truth meshes available. Therefore, we treat the bird category [32] as the major one for qualitative evaluation, through the task of keypoint transfer following previous work [18]. Given a pair of source and target images of two different object instances from a category, we map a set of annotated keypoints from the source image to the target image by first mapping them onto the learned shape template and then to the target image. Each mapping can be carried out by either the learned texture flow or the camera pose, as explained below.

To validate 3D reconstruction results, we also evaluate our model on rigid objects, e.g., cars [39], in terms of 3D IoU. However, we note that reconstruction of such rigid objects for which the ground truth 3D meshes/CAD models are easy to obtain, is not the major focus of this self-supervised method.

We first evaluate shape reconstruction on the bird category. Due to a lack of ground truth 3D shapes in the CUB-200-2011 dataset [32], we follow [15] and compute the mask reprojection accuracy – the intersection over union (IoU) between rendered and ground truth silhouettes. As shown in Table 1, our model is able to achieve comparable if not better mask reprojection accuracy compared to CMR [15], which unlike our method is learned with additional supervision from

Table 1: Quantitative evaluation of mask IoU and keypoint transfer (KT) on the CUB dataset [32]. The comparisons are against the baseline supervised models [15, 18].

(a) Metric	(b) CMR [15]	(c) CSM [18]	(d) Ours
Mask IoU ↑	0.706	-	0.734
KT (Camera) ↑	47.3	-	51.2
KT (Texture Flow) ↑	28.5	48.0	58.2

Table 2: Ablation studies of each proposed module by evaluating mask IoU and keypoint transfer (KT) on the CUB-200-2011 dataset [32].

(a) Metric	(b) Ours	(c) w/o L_{tcyc}	(d) w/o L_{sv} & L_{sp}	(e) with original [14]
Mask IoU ↑	0.734	0.731	0.744	0.731
KT (Camera) ↑	51.2	48.5	29.0	48.7
KT (Texture Flow) ↑	58.2	51.0	32.8	52.9

semantic keypoints. This indicates that our model is able to predict 3D mesh reconstructions and camera poses that are well matched to the 2D observations.

Next, we evaluate shape reconstruction on the car category. Although PASCAL3D+ [39] provides "ground truth" meshes (the most similar ones to the image in a mesh library), our reconstructed meshes are not aligned with these "ground truth" meshes since our self-suerpvised model is free to learn its own "canonical reference frame". Thus, to quantitatively evaluate the intersection over union (IoU) between the two meshes, following CMR [15], we exhaustively search a set of scale, translation and rotation parameters that best align to the "ground truth" meshes. Our method achieves an IoU (0.62) that is comparable to CMR [15] (0.64), even though the latter is trained with keypoints supervision.

Consider two different instances of a category as source and target images. To evaluate learned texture flow via keypoint transfer, given an annotated keypoint k^s in a source image (s), we map it to a triangle face (F_j) on the template using its learned flow I^s_{flow}. We then find all pixels (Ω_j) in the target image (t) that are mapped to the same triangle face F_j, by its texture flow I^t_{flow} and compute the geometric center of all pixels in Ω_j. We compare the location of the geometric center of Ω_j to the ground truth keypoint k^t and find the percentage of correct keypoints (PCK) as those that fall within a threshold distance $\alpha = 0.1$ of each other [18]. Figure 4 (a) in the appendix demonstrates qualitative visualizations of the keypoint transfer using texture flow and Table 1 shows that the texture flow learned by our method, even without supervision, outperforms the 2D→3D mappings learned by the supervised methods [15, 18].

To evaluate the learned camera pose via keypoint transfer, we first find the 3D template's vertex v that corresponds to a source image's annotated 2D keypoint k^s by rendering all 3D vertices using its predicted pose θ^s. Then, v is the vertex whose 2D projection lies closest to the keypoint k^s. Next, we render the point v with a target image's predicted pose θ^t and compare it to its ground truth keypoint k^t to compute PCK. Figure 4 (b) in the supplementary demonstrates the keypoint transfer results by the predicted camera pose. Table 1 shows that our model achieves favourable performance against the baseline method [15].

4.4 Ablation Studies

In this section, we discuss the contribution of each proposed module: (i) The semantic consistency constraint discussed in Section 3.1. (ii) The texture cycle consistency introduced in Section 3.3. We evaluate on the CUB-200-2011 dataset [32] and use the mask reprojection accuracy as well as the keypoint transfer (via texture flow and via camera pose) accuracy discussed in Section 4.3 as our metrics.

As shown in Table 2 (b) *vs.* (d) our baseline model trained without the semantic consistency constraint performs much worse at the keypoint transfer task than our full model, indicating this baseline model predicts incorrect texture flow and camera views. We note that this baseline model achieves better mask IoU because the model trained without any constraint is more prone to overfit to the 2D silhouette observations.

Our model trained without the texture cycle consistency constraint achieves worse performance (Table 2 (b) *vs.*(c)) at transferring keypoints using the predicted texture flow. This proves the effectiveness of the texture cycle consistency constraint in encouraging the model to learn better texture flow.

5 Failure Case and Limitations

Our method performs sub-optimally for objects with large concavities and objects with a genus greater than 0, such as horses and chairs. It captures the major shape characteristics of each instance but ignores some details, e.g., the two wings of flying birds, and the legs of zebras or horses are not separated, as shown in Figure 6 and Figure 7. Moreover, our method utilizes the SCOPS method to provide semantic part segmentation, and so it suffers when the semantic part segmentation is not accurate, as shown in the first row of Figure 8 in the supplementary or if the SCOPS model fails to discover meaningful parts for a certain category, such as airplanes, as shown in the supplementary document of [14]. We leave these failure cases and limitations to future works.

6 Conclusion

In this work, we learn a model to reconstruct 3D shape, texture and camera pose from single-view images, with only a category-specific collection of images and silhouettes as supervision. The self-supervised framework enforces semantic consistency between the reconstructed meshes and images and largely reduces ambiguities in the joint prediction of 3D shape and camera pose from 2D observations. It also creates a category-level template and a canonical semantic UV map, which capture the most representative shape characteristics and semantic parts of objects in each category, respectively. Experimental results demonstrate the efficacy of our proposed method in comparison to the state-of-the-art supervised category-specific reconstruction methods.

References

1. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. arXiv preprint arXiv:1512.03012 (2015) 3
2. Chen, W., Gao, J., Ling, H., Smith, E., Lehtinen, J., Jacobson, A., Fidler, S.: Learning to predict 3d objects with an interpolation-based differentiable renderer. In: NeurIPS (2019) 3
3. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV (2016) 2, 3
4. Collins, E., Achanta, R., Susstrunk, S.: Deep feature factorization for concept discovery. In: ECCV (2018) 4
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 10, 11
6. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: CVPR (2017) 3
7. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI (2009) 2
8. Girdhar, R., Fouhey, D., Rodriguez, M., Gupta, A.: Learning a predictable and generative vector representation for objects. In: ECCV (2016) 3
9. Gwak, J., Choy, C.B., Chandraker, M., Garg, A., Savarese, S.: Weakly supervised 3d reconstruction with adversarial constraint. In: 3DV (2017) 3
10. Häne, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3d object reconstruction. In: 3DV (2017) 3
11. Henderson, P., Ferrari, V.: Learning to generate and reconstruct 3d meshes with only 2d supervision. In: BMVC (2018) 2, 3
12. Henderson, P., Ferrari, V.: Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. IJCV (2019) 3
13. Honari, S., Molchanov, P., Tyree, S., Vincent, P., Pal, C., Kautz, J.: Improving landmark localization with semi-supervised learning. In: CVPR (2018) 4
14. Hung, W.C., Jampani, V., Liu, S., Molchanov, P., Yang, M.H., Kautz, J.: Scops: Self-supervised co-part segmentation. In: CVPR (2019) 2, 3, 4, 5, 6, 11, 13, 14
15. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: ECCV (2018) 2, 3, 4, 5, 8, 9, 11, 12, 13
16. Kato, H., Harada, T.: Learning view priors for single-view 3d reconstruction. In: CVPR (2019) 3, 9
17. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR (2018) 2, 3, 11
18. Kulkarni, N., Gupta, A., Tulsiani, S.: Canonical surface mapping via geometric cycle consistency. In: ICCV (2019) 2, 3, 4, 8, 9, 10, 12, 13
19. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv preprint arXiv:1811.00982 (2018) 10, 11
20. Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In: ICCV (2019) 3, 10
21. Liu, S., Saito, S., Chen, W., Li, H.: Learning to infer implicit surfaces without 3d supervision. In: NeurIPS (2019) 3
22. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: CVPR (2019) 3

23. Pan, J., Han, X., Chen, W., Tang, J., Jia, K.: Deep mesh reconstruction from single rgb images via topology modification networks. In: ICCV (2019) 3
24. Pepik, B., Gehler, P., Stark, M., Schiele, B.: 3d 2 pm–3d deformable part models. In: ECCV (2012) 2
25. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2016) 3
26. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017) 3
27. Rezende, D.J., Eslami, S.A., Mohamed, S., Battaglia, P., Jaderberg, M., Heess, N.: Unsupervised learning of 3d structure from images. In: NeurIPS (2016) 2
28. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: CVPR (2017) 4
29. Szabó, A., Favaro, P.: Unsupervised 3d shape learning from image collections in the wild. arXiv preprint arXiv:1811.10519 (2018) 3
30. Thewlis, J., Bilen, H., Vedaldi, A.: Unsupervised learning of object landmarks by factorized spatial embeddings. In: ICCV (2017) 4
31. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: CVPR (2017) 3
32. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011) 10, 12, 13, 14
33. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: ECCV (2018) 2, 3, 11
34. Wen, C., Zhang, Y., Li, Z., Fu, Y.: Pixel2mesh++: Multi-view 3d mesh generation via deformation. In: ICCV (2019) 2, 3
35. Wiles, O., Zisserman, A.: Silnet: Single-and multi-view reconstruction by learning from silhouettes. arXiv preprint arXiv:1711.07888 (2017) 2, 3
36. Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In: NeurIPS (2016) 3
37. Wu, S., Rupprecht, C., Vedaldi, A.: Photo-geometric autoencoding to learn 3d objects from unlabelled images. arXiv preprint arXiv:1906.01568 (2019) 3
38. Xiang, Y., Kim, W., Chen, W., Ji, J., Choy, C., Su, H., Mottaghi, R., Guibas, L., Savarese, S.: Objectnet3d: A large scale database for 3d object recognition. In: ECCV (2016) 3
39. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: WACV (2014) 10, 12, 13
40. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: NeurIPS (2016) 2, 3
41. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In: NeurIPS (2016) 11
42. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 9
43. Zhang, Y., Guo, Y., Jin, Y., Luo, Y., He, Z., Lee, H.: Unsupervised discovery of object landmarks as structural representations. In: CVPR (2018) 4
44. Zhu, J.Y., Zhang, Z., Zhang, C., Wu, J., Torralba, A., Tenenbaum, J., Freeman, W.: Visual object networks: Image generation with disentangled 3D representations. In: NeurIPS (2018) 3

45. Zhu, R., Kiani Galoogahi, H., Wang, C., Lucey, S.: Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In: ICCV (2017) 3