Supplementary material for 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

In this supplementary, we provide additional details, discussions, and experiments to support the original submission. We first discuss how to use the learned reconstruction model to improve the SCOPS [5] method in Section 1 and provide more implementation details of our model in Section 2. Then, we visualize the contribution of each module via ablation studies in Section 3. We further present more quantitative and qualitative results in Section 4 and Section 5.

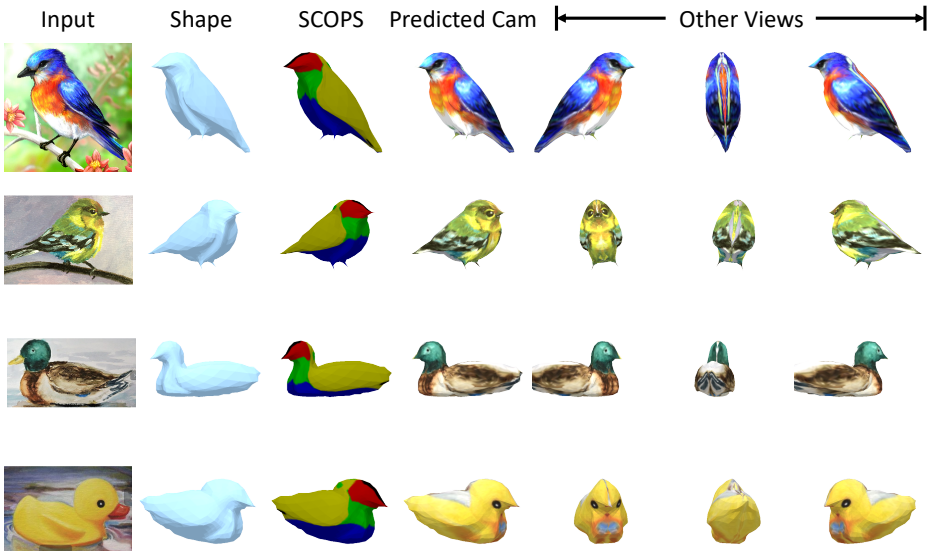

Fig. 1: Results of applying our reconstruction model on bird paintings.

1 Improving SCOPS by 3D Reconstruction

The proposed 3D reconstruction model can also be used to improve the learning of the self-supervised part segmentation model [5] (see Figure 5). The key

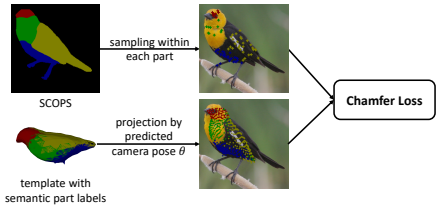

Fig. 2: **Visualization of the vertex-based semantic consistency constraint.** Points of the same color belong to the same semantic part.

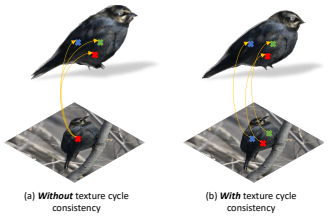

Fig. 3: **Visualization of the effectiveness of the texture cycle consistency constraint.**

intuition behind this is that the category-level canonical semantic UV map $\bar{P}_{\text{uv}}$ learned in Section 3.1 in the submission largely reduces noise in instance-based semantic UV maps. When combined with instance mesh reconstruction and the camera view, it provides reliable supervision for the SCOPS method.

By mapping the canonical UV map to the surface of each reconstructed mesh and rendering it from the predicted camera view, we obtain psuedo "ground truth" segmentation maps as supervision for training SCOPS. We use the semantic consistency constraint in Section 3.1 in the submission as a measurement to select the reliable reconstructions with high semantic consistency (i.e., with low probability and vertex-based semantic consistency loss values) to train SCOPS with. The improved SCOPS can, in turn, provide better regularization for our mesh reconstruction network, forming an iterative and collaborative learning loop.

In Figure 5, we visualize the results of improving SCOPS [5] with our 3D reconstruction network. Thanks to our learned canonical semantic UV map, the improved SCOPS method is able to predict the correct parts and accurately localizes them with a more precise size (head and neck parts in column 1,2,3).

Quantifying the improved SCOPS method numerically is non-trivial as the ground-truth segmentation labels for the parts are not available in all the dataset that we use [14, 16, 1, 11]. Instead, we indirectly measure its improvement by training two models, each of which uses the semantic part segmentation predicted either by the original or the improved SCOPS method. As shown in Table 2 (b) *vs.*(e) in the main paper, our keypoint transfer performance drops by 5.3% and 2.5% via texture flow and camera pose if we use the original SCOPS model. More qualitative visualizations of the improvement of SCOPS can be found in the appendix.

2 Implementation Details

2.1 Selective Aggregation in the M-step

Computing Category-level Template In the M-step (Section 3.2 of the submission), we update the template by decoding the averaged shape feature via the shape decoder. Instead of using all training samples to obtain the averaged

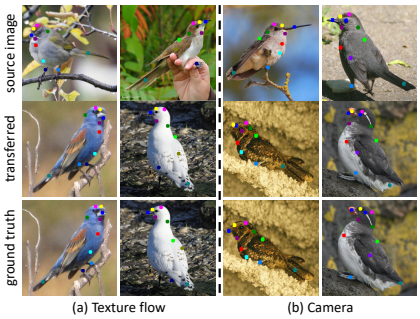

Fig. 4: **Qualitative visualization of keypoint transfer.** We show comparisons against the ground truth keypoints in each column.

Fig. 5: **Improvements to the SCOPS [5] method.** Notice the more consistent size and shape of part segments with the improved method.

feature, we select a subset of the training samples to form a set $\mathcal{Q}$ and compute the averaged feature for the samples in this set. In the following, we explain why and how to form this set $\mathcal{Q}$ used in Eq.(4) of the submission. Empirically we found that for several categories, there exist ambiguities that produce inconsistent mesh reconstructions, e.g., side-view images of horses could be reconstructed with their heads on either the left or the right side. Aggregating such instance meshes leads to incorrect estimation of the category-level template. To resolve this, we select a subset of the reconstructed meshes whose viewpoints roughly match (e.g., horses with heads on the left side). To do so, from the meshes reconstructed for all the training images, we first choose an instance with the most "reliable" reconstruction results, i.e., the instance whose rendered silhouette have the largest intersection over union (IoU) with its corresponding ground truth silhouette, as an exemplar (e.g., a horse shape with its head on the left). We then use the top k training samples with meshes that are most similar to that of the exemplar mesh to form the subset $\mathcal{Q}$ in Eq.(4) (e.g., all chosen horse samples have heads on the left). We measure the similarity between an individual instance mesh and the exemplar mesh by computing the IoU between their rendered silhouettes.

Computing Canonical Semantic UV Map Similarly, when we update the canonical semantic UV map using Eq.(1) (see Section 3.2 in the submission), to avoid using training samples with outliers, e.g., those caused by inaccurate prediction of I^i_{flow}, we choose an exemplar training example with the smallest perceptual distance objective (see Section 3.2 in the submission), and form the set $\mathcal{U}$ of the top k training samples that have the most similar semantic UV maps (as measured by the L2 norm) to the exemplar.

2.2 Network Architecture and Other Objectives

Network Architecture We present the details of our network architecture as well as training objectives in Figure 6. We use the same network as in CMR [7],

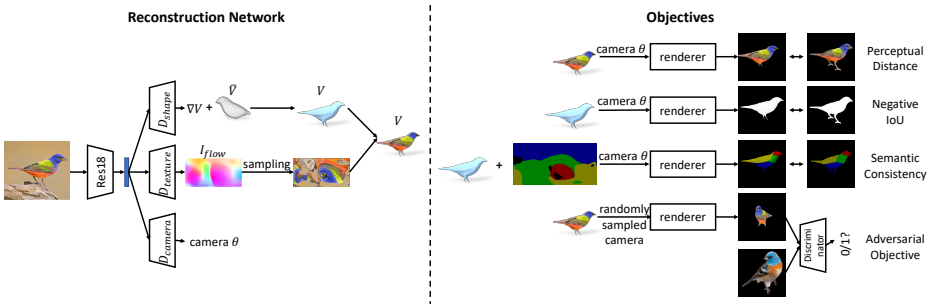

Fig. 6: Network Architecture and Objectives.

where: (i) The encoder is the ResNet18 network [4] with four residual blocks and is pretrained on the ImageNet [1] dataset. (ii) The shape decoder consists of one fully connected layer to decode shape deformation ΔV. (iii) The texture decoder contains two fully-connected layers followed by eleven upsampling and convolution layers to predict the texture flow I_{flow}. (iv) The camera pose decoder contains three parallel fully connected layers to predict the scale, translation and rotation, and these three parameters together compose the camera pose θ. Note that we use the one camera hypothesis in the first EM training round and use multiple camera hypotheses (eight camera hypotheses) as in [10, 6, 13] to avoid local minima in the subsequent rounds. To render the reconstructed meshes, we utilize the Soft Rasterizer [12] instead of the Neural Mesh Renderer [9] used in the CMR [7]. This is because it provides the probability map described in Section 3.3 for the texture cycle consistency constraint.

Smoothness Term In addition to the objectives discussed in Section 3.2 of the submission, we further utilize a graph Laplacian constraint to encourage the reconstructed mesh surface to be smooth [7, 12], and adopt an edge regularization to penalize irregularly-sized faces as in [15, 2]. More details can be found in [7, 12, 15, 2].

Adversarial Training To constrain the reconstructed meshes to look plausible from all views, we also introduce adversarial training [3] into the mesh reconstruction framework [8]. We render the reconstructed mesh from a randomly sampled camera pose to obtain an image I_{rd}, and pass it together with a random real image I_{rl} into a discriminator. By learning to discriminate between the real and rendered images, the discriminator learns shape priors and constrains the reconstruction model to generate meshes that are plausible from all viewpoints. The adversarial loss is:

$$L_{\text{adv}}(R, D) = \mathbb{E}_{I_{\text{rl}}}[\log D(I_{\text{rl}})] + \mathbb{E}_{I_{\text{rd}}}[\log(1 - D(I_{\text{rd}}))], \tag{1}$$

where R and D are the reconstruction and discriminator networks, respectively. Figure 6 illustrates the adversarial objective.

Table 1: Settings of each baseline models in Section 3.1.

Module	category-level template	semantic consistency	adversarial training
baseline (a)	×	×	×
baseline (b)	✓	×	×
baseline (c)	✓	✓	×
Ours	✓	✓	✓

2.3 Network Training

We train the reconstruction network with an initial learning rate of $1e-4$ and gradually decay it by a factor of 0.5 every 2000 iterations. The network is trained for two EM training rounds (each training round contains one E and one M-step) on four NVIDIA Tesla V100 GPUs for two days. We found that two rounds of EM training are sufficient to generate high-quality reconstruction results. During the inference stage, the model takes 0.022 seconds to reconstruct a 3D mesh from a 256×256 sized single-view image on a single NVIDIA Tesla V100 GPU. In Figure 7, we show the learned template shape as well as the semantic parts after the first (left figure) and second M-steps (right figure), where both the template shape and the semantic parts after the second M-step are better than the first.

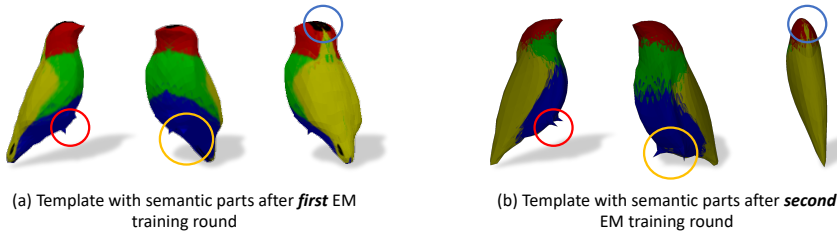

Fig. 7: Visualization of the learned template and semantic parts. Notice the improvements of the template after the second M-step compared to the first, i.e., better feet shape in the red and yellow circles, and a part of the head (blue circle) that was mistakenly assigned to the background (colored in black) in the first step is corrected (colored in red) in the second M-step.

3 Ablation Studies

3.1 Ablation Studies for Different Objectives

We show the results of three baselines in Figure 10. The experimental settings for each are illustrated in Table 1 and are the following: (a) a basic model trained with only the texture cycle consistency constraint described in Section 3.3 of the submission, but without any other proposed modules, i.e., the category-level template, the semantic consistency constraint and the adversarial training; (b)

learning the model in (a) together with the category-level template; and (c) learning the model in (b) with the additional semantic consistency constraint.

As shown in Figure 10, the basic model (a) reconstructs meshes that only appear plausible from the observed view to match the 2D supervision (images and silhouettes). It fails to generate plausible results for unobserved views, e.g., for all the 3 examples. On adding template shape learning (Section 3.2 in the submission) to (a), the model in (b) learns more plausible reconstruction results across different views. This is because it is easier for the model to learn residuals w.r.t a category-level template compared to w.r.t a sphere, to match the 2D observations. However, without semantic part information, the model still suffers from the "camera-shape ambiguity" discussed in Section 1 of the manuscript. For instance, the head of the template is deformed to form the tail and the wing's tip in the first and second examples, respectively in Figure 10. By additionally including the semantic consistency constraint in the model (c), the network is able to reduce the "camera-shape ambiguity" and predict the correct camera pose as well as the correct shape. Furthermore, adding adversarial training introduces better reconstruction details, as shown in Figure 10 (d). For instance, the bird may have more than two feet without the adversarial training constraint as demonstrated in the third example in Figure 10.

In addition, we demonstrate the effectiveness of the texture flow consistency constraint by visualizing the keypoint transfer results in Figure 11. The model trained without this constraint performs worse than our full model, especially when the bird has a uniform color, e.g., the second and the last examples in Figure 11. Figure 11 also shows that the proposed method performs favourably against the baseline CSM [10] method.

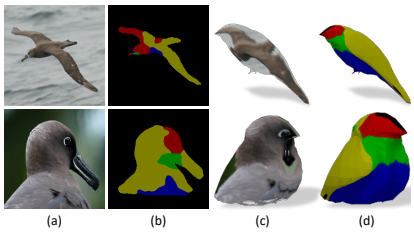

Fig. 8: Failure cases. (a) Input images. (b) Semantic part segmentations predicted by the SCOPS method. (c) Reconstructed meshes. (d) Reconstructed meshes with the canonical semantic UV map.

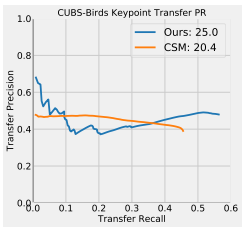

Fig. 9: Keypoint Transfer PR Curves. The legend of the plot represents the area under the curve, our method achieves an APK of 25.0, which is better than the baseline method [10].

Table 2: Ablation studies of the probability and vertex-based semantic consistency constraints by evaluating the mask IoU and the keypoint transfer (KT) task on the CUB-200-2011 dataset [14].

(a) Metric	(b) Ours	(c) w/o L_{sv}	(d) w/o L_{sp} original [5]
Mask IoU ↑	0.734	0.6069	0.6418
KT (Camera) ↑	51.2	30.7	51.0
KT (Texture Flow) ↑	58.2	29.5	53.3

3.2 Ablation Studies on Semantic Consistency Constraints

We show an ablation study of the probability and vertex-based semantic consistency constraints in Table 2, where both constraints contribute to the reconstruction network.

4 More Quantitative Evaluations

4.1 APK Evaluation

For the keypoint transfer task, in Figure 9, we demonstrate the precision versus recall curves of our method (via texture flow) and of the CSM [10] method on the CUB-200-2011 [14] *test* dataset. Our method, even without the template prior, outperforms the baseline CSM [10] method in terms of the Keypoint Transfer AP metric (APK, $\alpha = 0.1$).

5 More Qualitative Evaluations

We show more qualitative results for birds in Figure 12. We also show one application of our model to reconstruct 3D meshes of 2D bird paintings in Figure 1. Reconstruction of rigid objects (cars and motorbikes) is demonstrated in Figure 14, horses and cows in Figure 13, and penguins and zebras in Figure 15. Note that we use six semantic parts for the car category to encourage the SCOPS method [5] to differentiate between the front and side of cars. For other objects, we use four semantic parts.

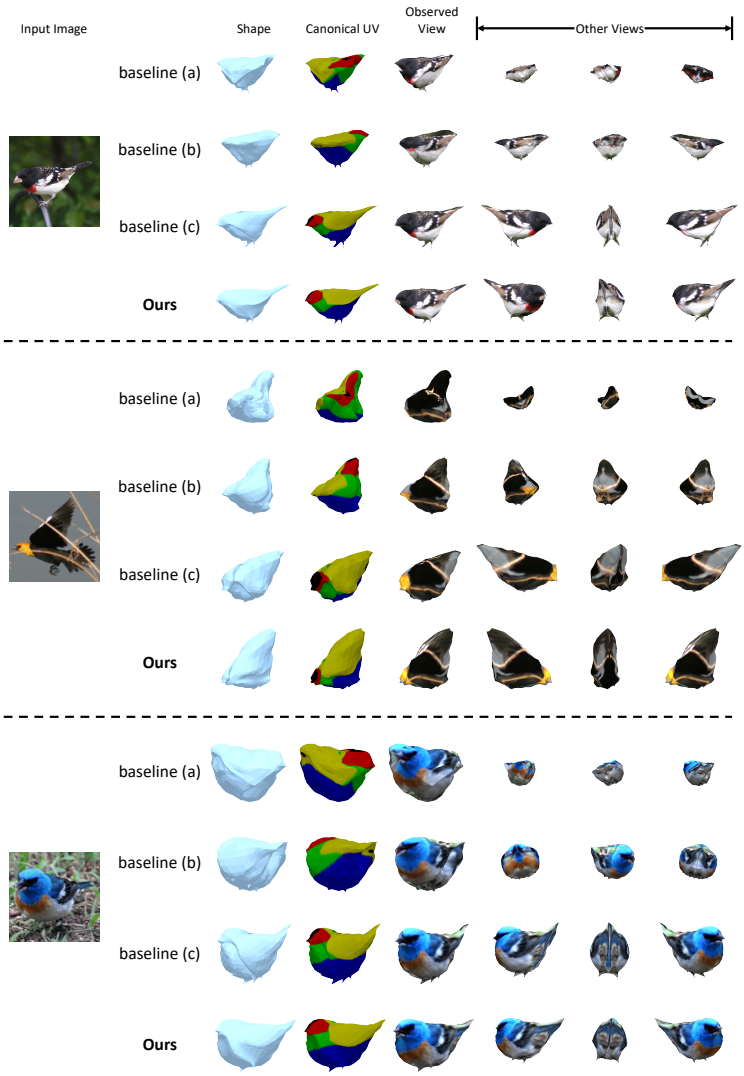

Fig. 10: Visualization of the contribution of each module. The settings of baselines (a), (b), (c) can be found in Table 1

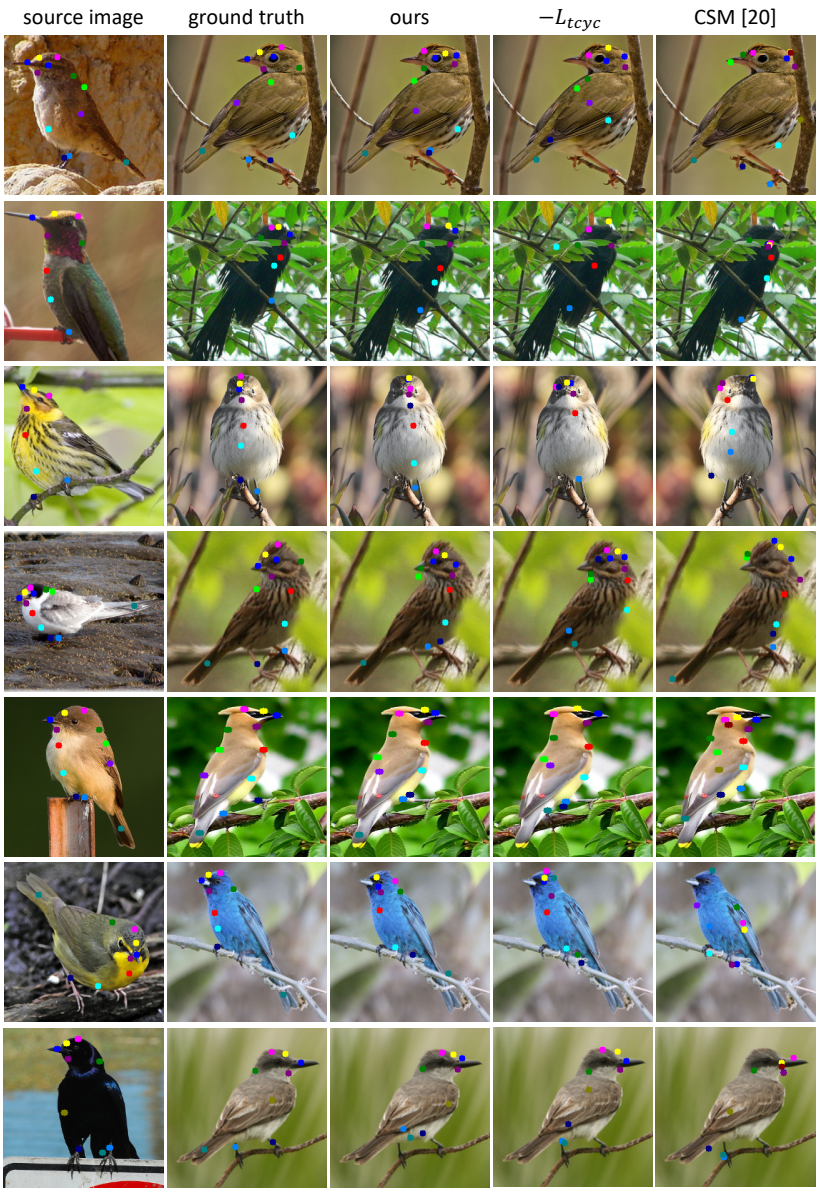

Fig. 11: Visualization of keypoint transfer using texture flow.

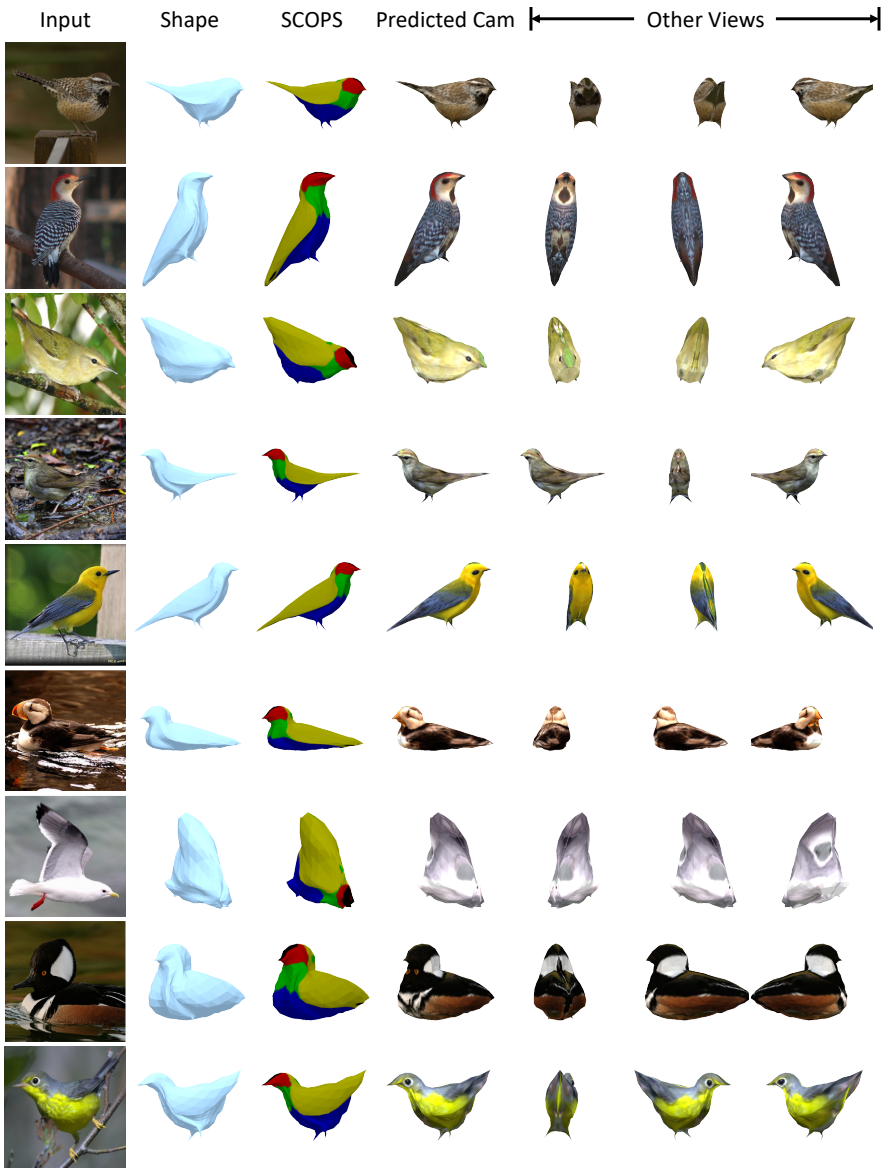

Fig. 12: More qualitative results of birds.

Fig. 13: More qualitative results of horses and cows.

Fig. 14: More qualitative results of motorbikes and cars.

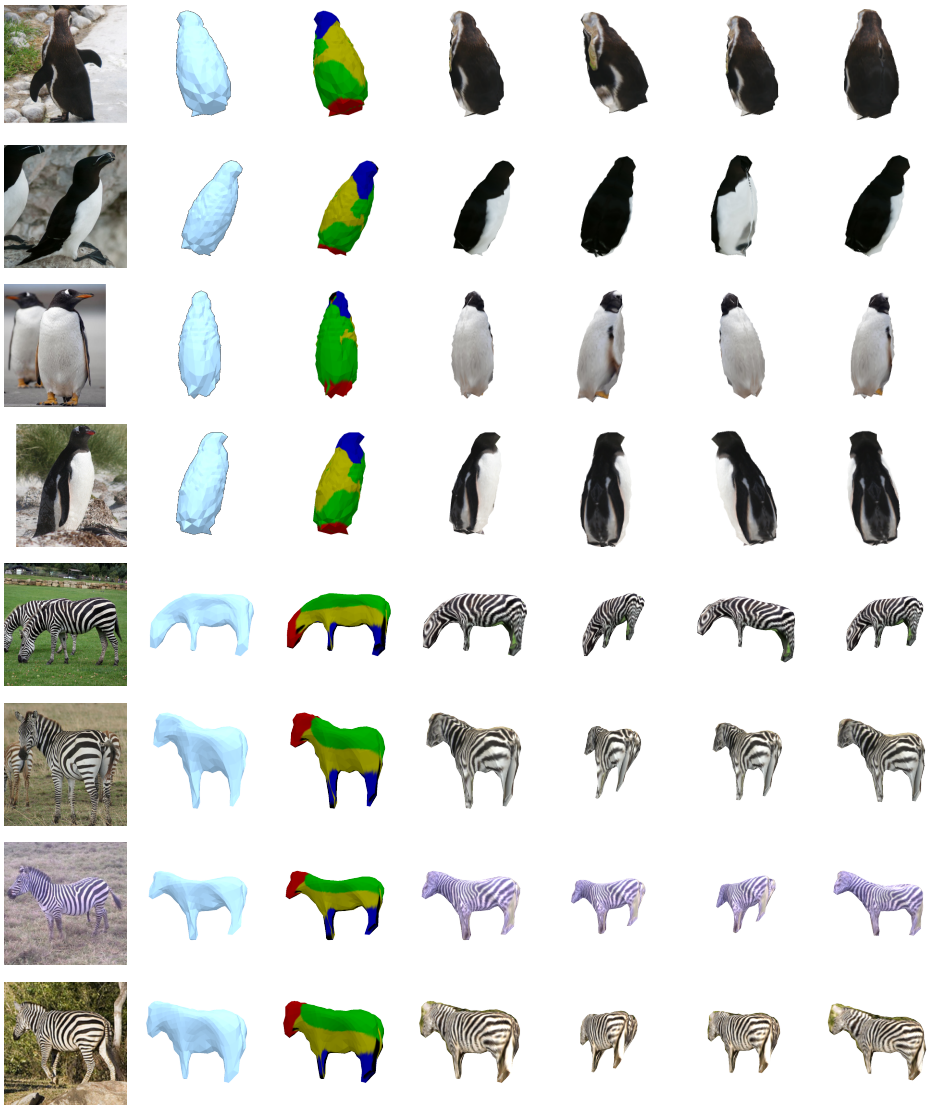

Fig. 15: More qualitative results of zebras and penguins.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009) 2, 4
2. Gkioxari, G., Malik, J., Johnson, J.: Mesh r-cnn. ICCV (2019) 4
3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) 4
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 4
5. Hung, W.C., Jampani, V., Liu, S., Molchanov, P., Yang, M.H., Kautz, J.: Scops: Self-supervised co-part segmentation. In: CVPR (2019) 1, 2, 3, 7
6. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: NeurIPS (2018) 4
7. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: ECCV (2018) 3, 4
8. Kato, H., Harada, T.: Learning view priors for single-view 3d reconstruction. In: CVPR (2019) 4
9. Kato, H., Ushiku, Y., Harada, T.: Neural 3d mesh renderer. In: CVPR (2018) 4
10. Kulkarni, N., Gupta, A., Tulsiani, S.: Canonical surface mapping via geometric cycle consistency. In: ICCV (2019) 4, 6, 7
11. 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) 2
12. Liu, S., Li, T., Chen, W., Li, H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In: ICCV (2019) 4
13. Tulsiani, S., Efros, A.A., Malik, J.: Multi-view consistency as supervisory signal for learning shape and pose prediction. In: CVPR (2018) 4
14. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011) 2, 7
15. 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) 4
16. Xiang, Y., Mottaghi, R., Savarese, S.: Beyond pascal: A benchmark for 3d object detection in the wild. In: WACV (2014) 2