Supplemental Material of Discovering 3D Parts from Image Collections

Chun-Han Yao
UC Merced

Wei-Chih Hung
Waymo

Varun Jampani
Google

Ming-Hsuan Yang
UC Merced

1. Overview

In this document, we present the implementation details, model analysis, and additional results of our method. We also provide a short video to explain our framework with illustrations and visual results.

2. Implementation Details

We implement our framework with PyTorch [8] and train the models on TITAN X GPUs. The Part-VAE, reconstruction model, and discriminator are parametrized with neural networks. We apply Batch Normalization [3] after each convolutional layer and use ReLU [1] as the activation function of the middle layers.

2.1. Shape Reconstruction

Given an input image, the reconstruction model produces a 3D centroid and shape encoding for each object part. The network architecture of the image encoder and shape decoder are illustrated in Figure 1 and Figure 2, respectively. To obtain a part shape, we pass individual encodings through the decoder of Part-VAE and apply the predicted deformations to a template mesh. We use a spherical mesh with $N_v = 642$ and $N_f = 1280$ as template in our experiments. Finally, the whole object shape is composited by shifting the deformed meshes to the predicted centroids and combining their vertices and faces. Note that we do not penalize part overlap with any separation loss. Instead, the goal to cover the entire object silhouette with the proposed part prior enables our model to discover the best part configurations automatically.

2.2. Color Reconstruction

Given the extracted image features, the decoder predicts a texture image for each object part. We then obtain the surface colors for each part mesh by a standard UV mapping predefined on a template mesh. In our experiments, we set the size of texture image $(H_t, W_t) = (64, 64)$ and the texture resolution for each mesh surface $N_t = 2$ We show the detail mechanism and network architecture in Figure 2.

2.3. Part-VAE

We show the network architecture of the Part-VAE in Figure 3. To train the Part-VAE, we generate primitive meshes like cuboids, ellipsoids, cylinders, and cones using the Open3D [11] library. The meshes are centered at the origin but generated with random parameters, e.g. width, height, and rotation.

2.4. Discriminator

To perform class conditioning, we convert the input class labels into one-hot vectors and use it to condition the last (fully-connected) layer of the discriminator. Detail architecture is shown in Figure 4.

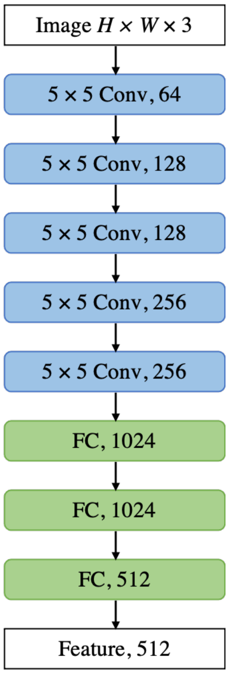

Figure 1: **Network architecture of the image encoder.** The extracted features are then passed through the decoder.

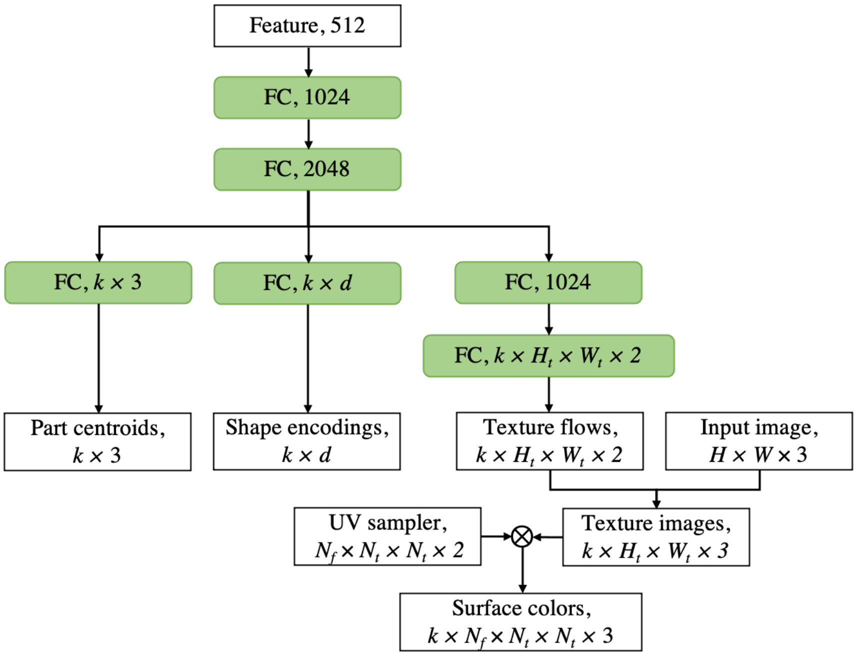
Feature, 512
FC, 1024
FC, 2048
FC, $k \times 3$
FC, $k \times d$
FC, 1024
FC, $k \times H_t \times W_t \times 2$
Part centroids, $k \times 3$
Shape encodings, $k \times d$
Texture flows, $k \times H_t \times W_t \times 2$
Input image, $H \times W \times 3$
UV sampler, $N_f \times N_t \times N_t \times 2$
$\otimes$
Texture images, $k \times H_t \times W_t \times 3$
Surface colors, $k \times N_f \times N_t \times N_t \times 3$

Figure 2: **Network architecture of the shape and color decoder.** k is the number of parts, (H_t, W_t) is the size of texture image, N_t is the texture resolution of each mesh surface, and $\otimes$ denotes the UV sampling operation.

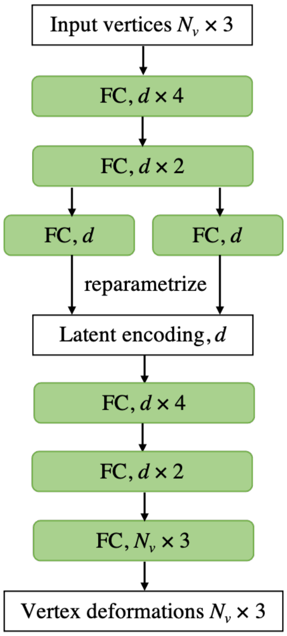

Figure 3: **Network architecture of the proposed Part-VAE.** N_v denotes the number of mesh vertices and d is the latent dimension. The deformations are applied on a template spherical mesh.

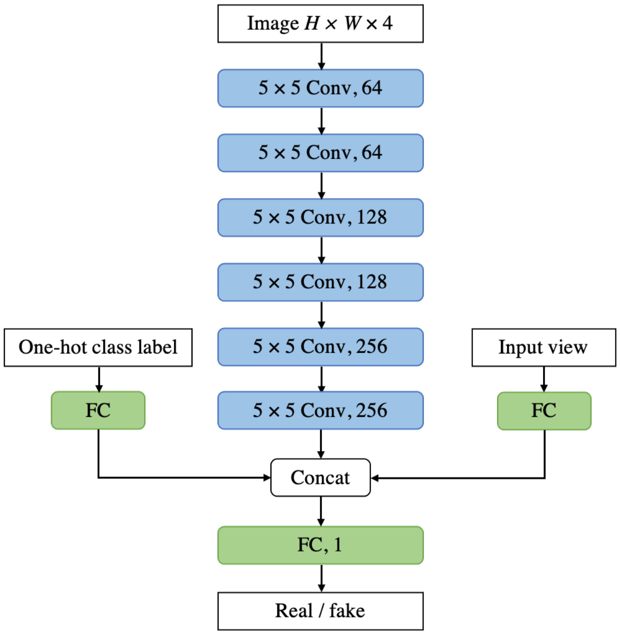

Figure 4: **Network architecture of the discriminator for part and view adversarial learning.** $\oplus$ and $\otimes$ denote element-wise sum and multiplication, respectively.

3. Additional Analysis

3.1. Detailed results on ShapeNet

We show the detailed voxel IoU results of all 13 ShapeNet [2] classes in Table 1. Note that the results in SoftRas [6] paper are reported on a different ShapeNet dataset using 24-view training, and the results here are given by our implementation.

Table 1: **Ablative evaluations on the ShapeNet dataset [2].** The base model reconstructs an object shape with 3 meshes, each is fully-deformable as in SoftRas [6] (PP: part prior, VA: view adversarial learning, PA: part adversarial learning, CR: color reconstruction).

Method	Airplane	Bench	Dresser	Car	Chair	Display	Lamp	Speaker	Rifle	Sofa	Table	Phone	Vessel	All
SoftRas [6]	52.2	32.4	54.2	65.7	40.4	32.8	44.4	57.9	48.8	48.5	39.3	40.7	51.5	46.9
VPL [4]	53.1	**38.5**	59.1	70.1	45.4	**42.3**	44.1	57.0	52.1	**50.8**	**44.4**	**60.1**	49.8	51.3
LPD (ours)	**57.3**	37.3	**60.9**	**75.2**	**45.5**	40.8	**49.6**	**63.3**	**54.5**	50.1	44.3	52.7	**56.2**	**52.9**

3.2. Ablation studies on Part-VAE

We perform ablation studies on the training strategies of Part-VAE to justify our model design. First, we use VAE instead of a vanilla auto-encoder for latent part encoding to create a continuous latent space. It not only improves the reconstruction accuracy (vanilla auto-encoder: 51.7, VAE: 52.9 on ShapeNet), but allows us to easily generate realistic shapes by interpolation or random sampling in the latent space. In Table 2, we show the results of different Part-VAE training methods. We also show the ablative evaluations on latent dimension in Table 3.

Table 2: **Ablations studies on Part-VAE training in the ShapeNet experiment.** The results show that primitive regularization is needed to achieve a fine balance between geometric simplicity and shape faithfulness of each part in both the pre-training and fine-tuning stages.

Pre-training	Fine-tuning	Primitive Regularization	Voxel IoU
✓			40.9
	✓	✓	50.1
✓	✓		51.3
✓	✓	✓	**52.9**

Table 3: **Ablation studies on the latent dimension (d) of Part-VAE.** We report the voxel IoU results on ShapeNet. The regularization might not be sufficient with a large d, while a small d would result in worse reconstructions due to insufficient capacity. The results show that our model with $d = 64$ achieves a better trade-off between the degree of deformation and shape regularization.

Latent dimension	Airplane	Car	Chair	All 13 classes
$d = 256$	57.0	72.8	44.8	52.0
$d = 128$	**57.3**	73.6	45.3	52.8
$d = 64$	**57.3**	**75.2**	**45.5**	**52.9**
$d = 32$	56.8	74.0	42.8	52.5
$d = 16$	55.8	72.5	41.5	51.8

3.3. Multi-view training

We propose the part-based method aiming to deal with a weakly-supervised setting (1-view training). However, our method can be easily extended to multi-view training by rendering the reconstructed mesh from different viewpoints and calculating the silhouette loss as a stronger supervision. As a reference, we show the quantitative results of multi-view training in Table 4.

3.4. Latent Part Analysis

To analyze the distribution of discovered object latent part encodings, we visualize them using t-SNE [7] plots. We show the t-SNE plots of latent part encodings and reconstructed mesh vertices in Figure 5 and Figure 6, respectively. As demonstrated in the figures, the discovered object parts are clearly separated in both the latent space and the 3D output space.

Table 4: **Ablations studies on multi-view training in the ShapeNet experiment.** We report the whole-object voxel IoU using 1 view, 2 views, or 20 views per object for training. Our part-based method achieves higher reconstruction accuracy compared to the existing methods, especially in a weakly-supervised setting (1 view or 2 views).

Method	1 view	2 views	20 views
SoftRas [6]	46.9	55.5	64.6
VPL [4]	51.3	58.3	65.5
LPD (ours)	**52.9**	**60.1**	**66.3**

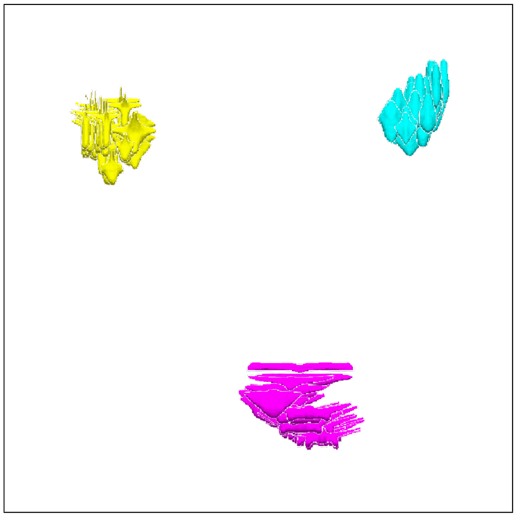

Figure 5: **t-SNE plot of the latent encoding of airplane parts.**

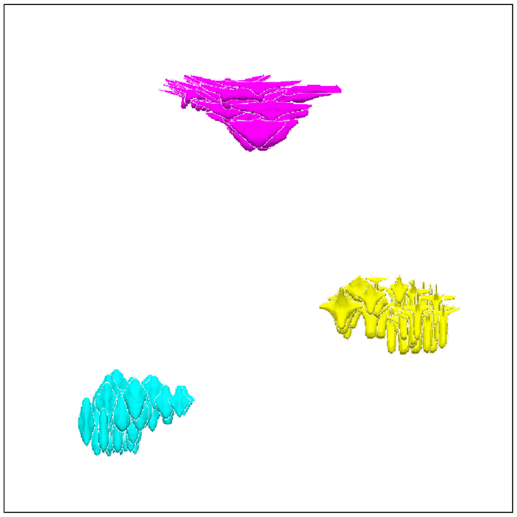

Figure 6: **t-SNE plot of the mesh vertices of airplane parts.**

3.5. Failure Cases

In Figure 7, we show some failure cases produced by our model. For object shapes that are compact and primitive-like by itself, our model tends to produce highly-overlapping parts. On the other hand, if an object is composed of multiple parts connected by a thin structure, our model is likely to ignore the thin connection and result in separated parts. We have experimented with various part separation and connection constraints to deal with such cases. However, we find that automatic part discovery is more general and achieves the highest reconstruction accuracy on average.

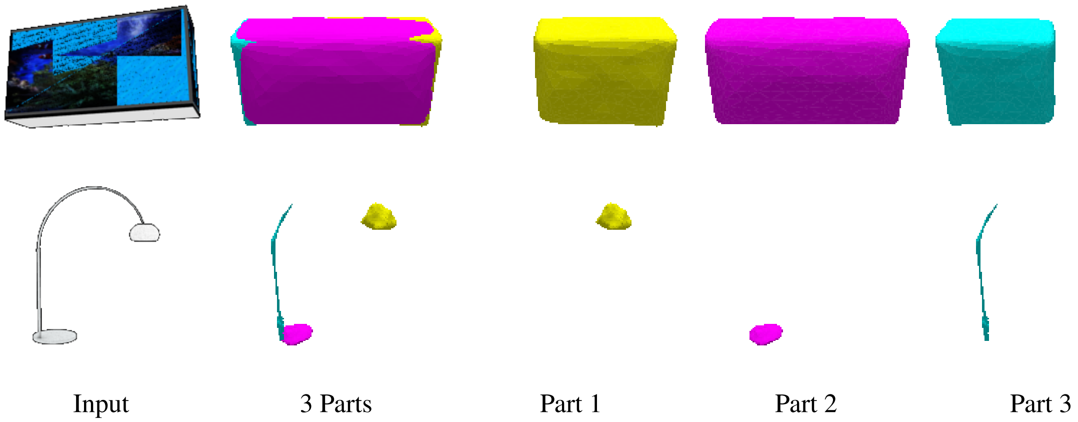

Figure 7: **Failure cases.** Our model is likely to produce highly-overlapping parts when the object is compact and primitive-like as a whole. If the object has a thin structure that is barely captured by the object silhouette, our model may produce disconnected parts.

4. Additional Results

We present more qualitative results on the ShapeNet [2] dataset. In Figure 8 and 10, we show the qualitative comparisons of our approach and the baseline methods. The results show that LPD (ours) discovers more meaningful and semantically consistent parts compared to using primitive part shapes or without any part prior. We also show some example 6-part reconstructions in Figure 9 and 11. More 3-part reconstruction results of diverse ShapeNet classes are shown in Figure 12 and Figure 13. The qualitative results demonstrate that the discovered parts are meaningful, consistent across object instances, and general across object classes.

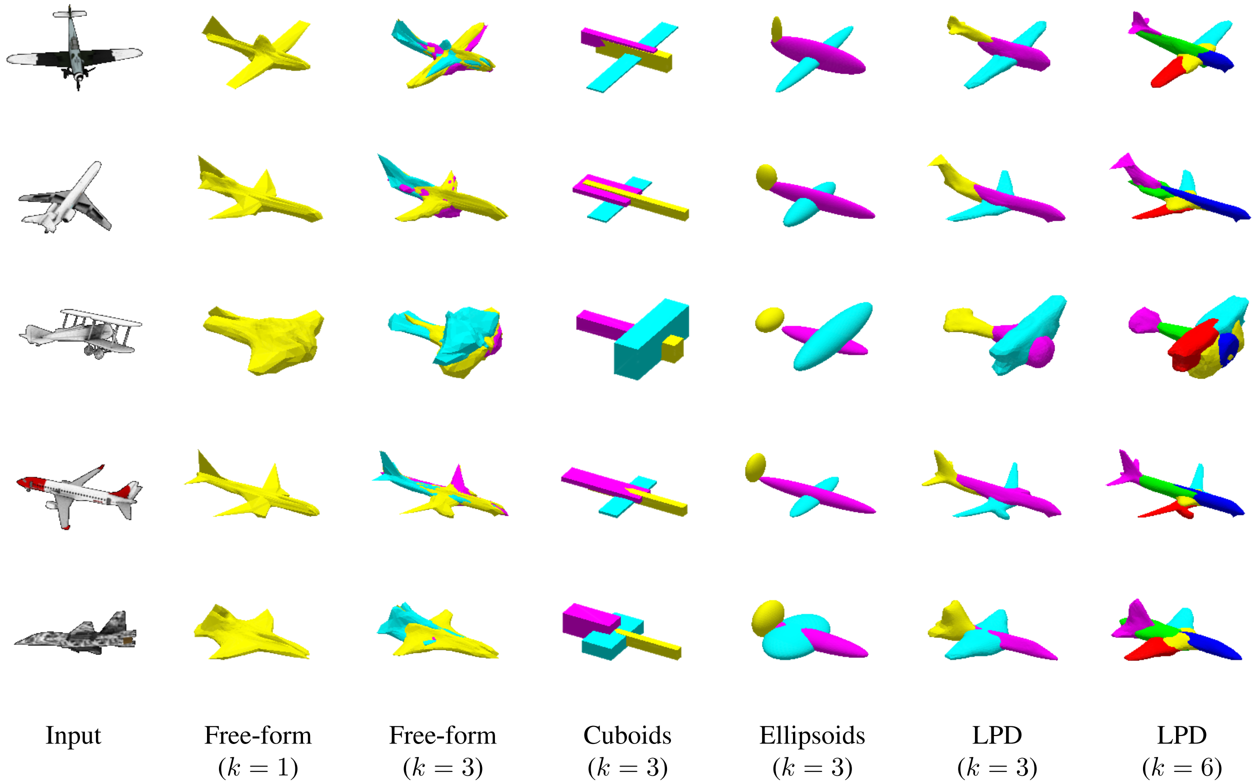

Figure 8: **Qualitative comparisons on the ShapeNet airplane examples.** We show sample results by different methods, including 3 part reconstruction with free-form meshes, cuboids, or ellipsoids, and LPD (ours) with 3 and 6 parts. Our discovered parts are consistent across diverse object instances and capture the detail structures faithfully.

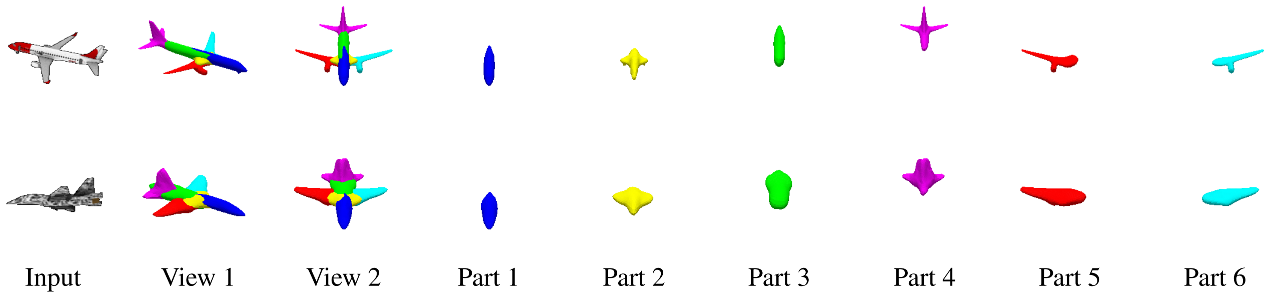

Figure 9: **Qualitative results of 6-part reconstruction.** We show the 6-part airplane reconstructions rendered from two different viewpoints. The results show that the discovered object parts are meaningful and consistent across instances.

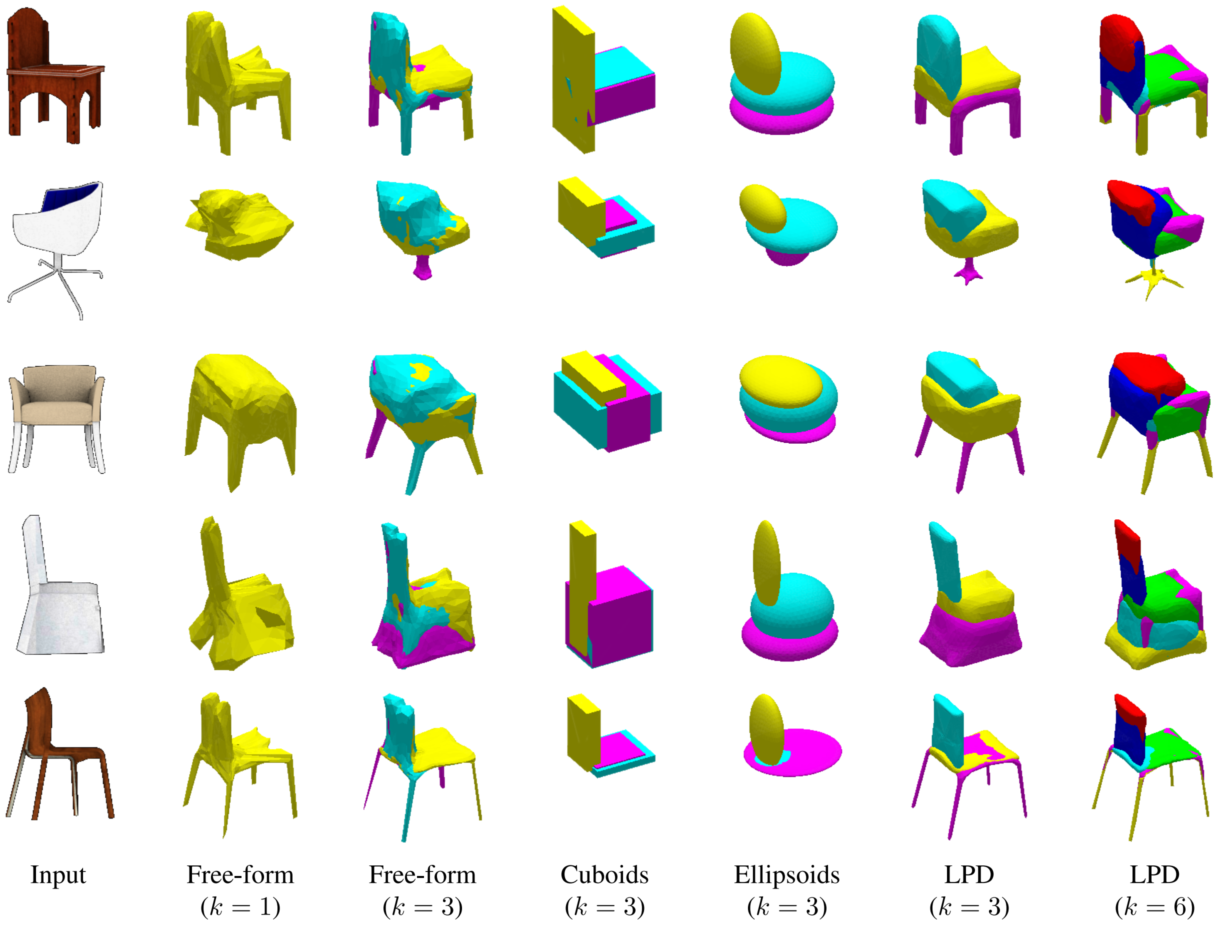

Figure 10: **Qualitative comparisons on the ShapeNet chair examples.** We show sample results by different methods, including 3 part reconstruction with no part prior, cuboids, or ellipsoids, and LPD (ours) with 3 and 6 parts. Our discovered parts are consistent across diverse object instances and capture the detail structures faithfully.

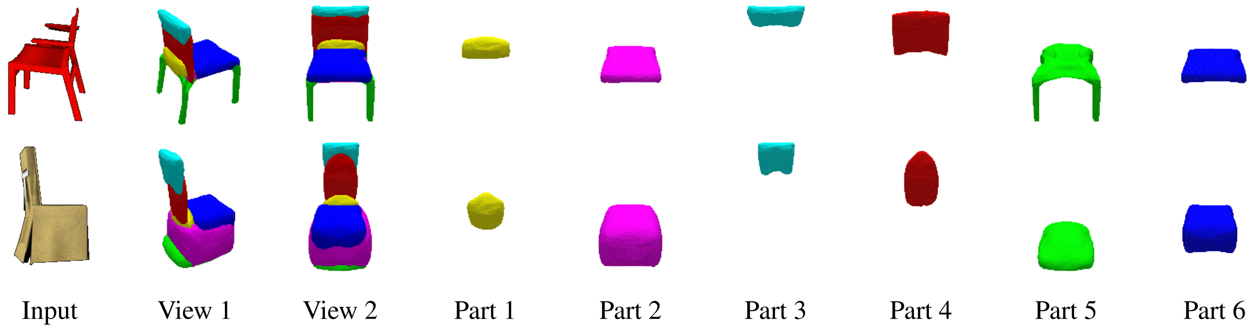

Figure 11: **Qualitative results of 6-part reconstruction.** We show the 6-part airplane reconstructions rendered from two different viewpoints. The results show that the discovered object parts are meaningful and consistent across instances.

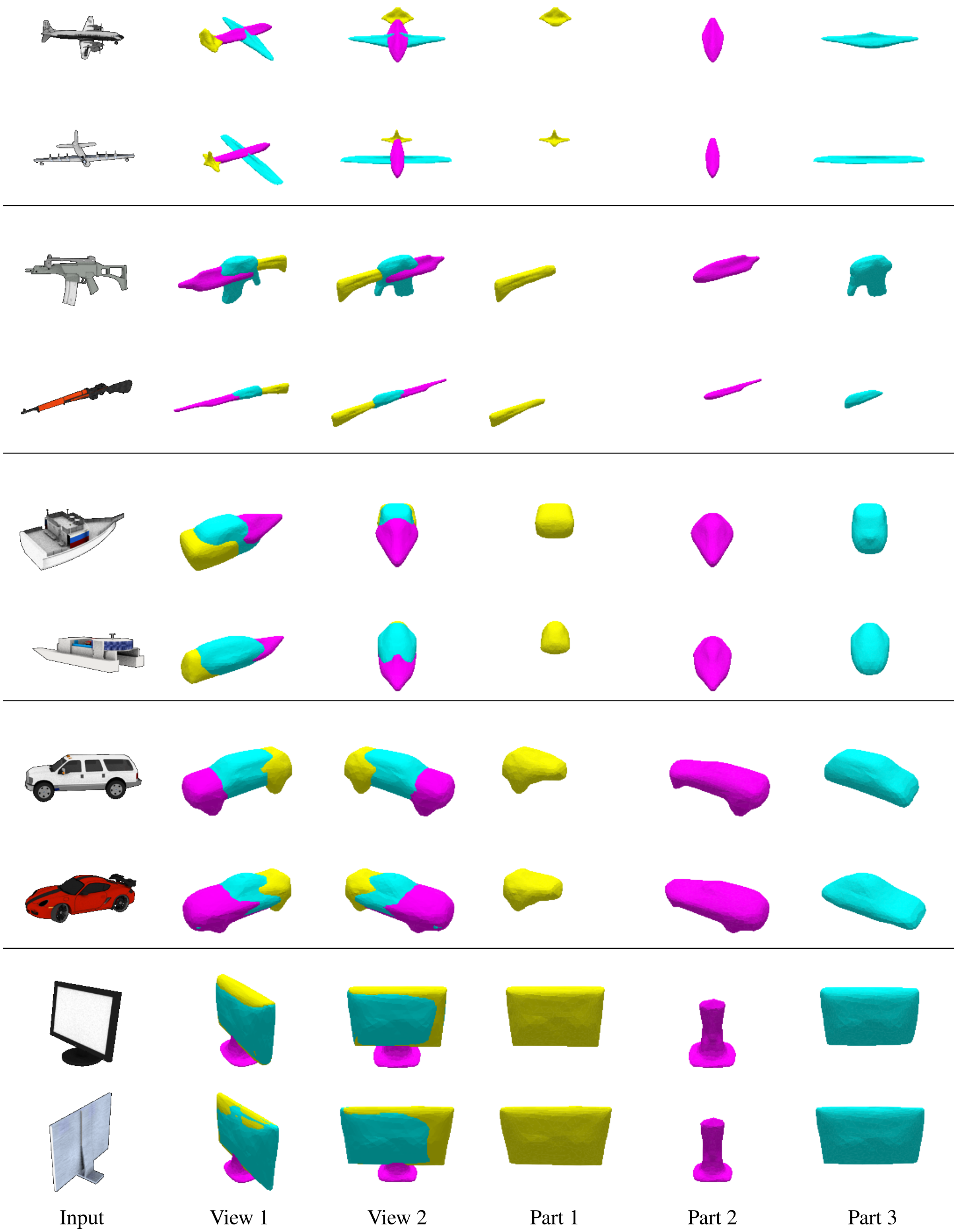

Figure 12: **Qualitative results on the ShapeNet dataset.** Our models adopt the proposed Part-VAE and adversarial learning. We show the reconstruction results of airplanes, rifles, watercrafts, cars, and displays (from top to bottom). Each sample is rendered from two different viewpoints. The results show that the discovered object parts are meaningful and consistent across instances.

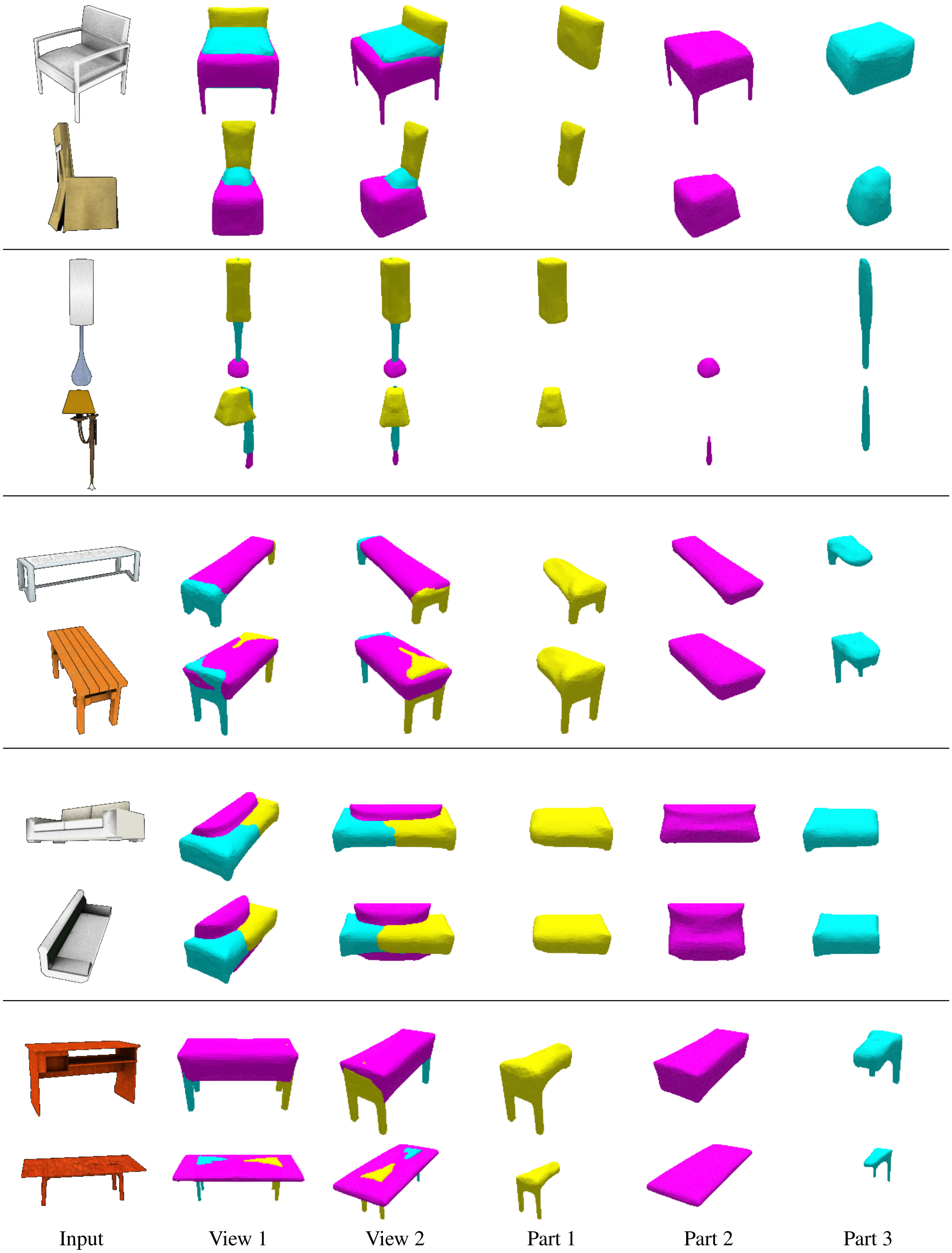

Figure 13: **Qualitative results on the ShapeNet dataset.** Our models adopt the proposed Part-VAE and adversarial learning. We show the reconstruction results of airplanes, rifles, watercrafts, cars, and displays (from top to bottom). Each sample is rendered from two different viewpoints. The results show that the discovered object parts are meaningful and consistent across instances.

4.1. Additional Results on Pascal 3D+

We show more qualitative results on the Pascal 3D+ dataset [10] in Figure 14. Note that this dataset also contains the images from ImageNet [9] and the silhouette and viewpoint annotations estimated by a self-supervised method [5]. Despite the noisy ground-truth annotations and complicated object shapes in real-world scenes, our method still discovers meaningful and semantically consistent parts.

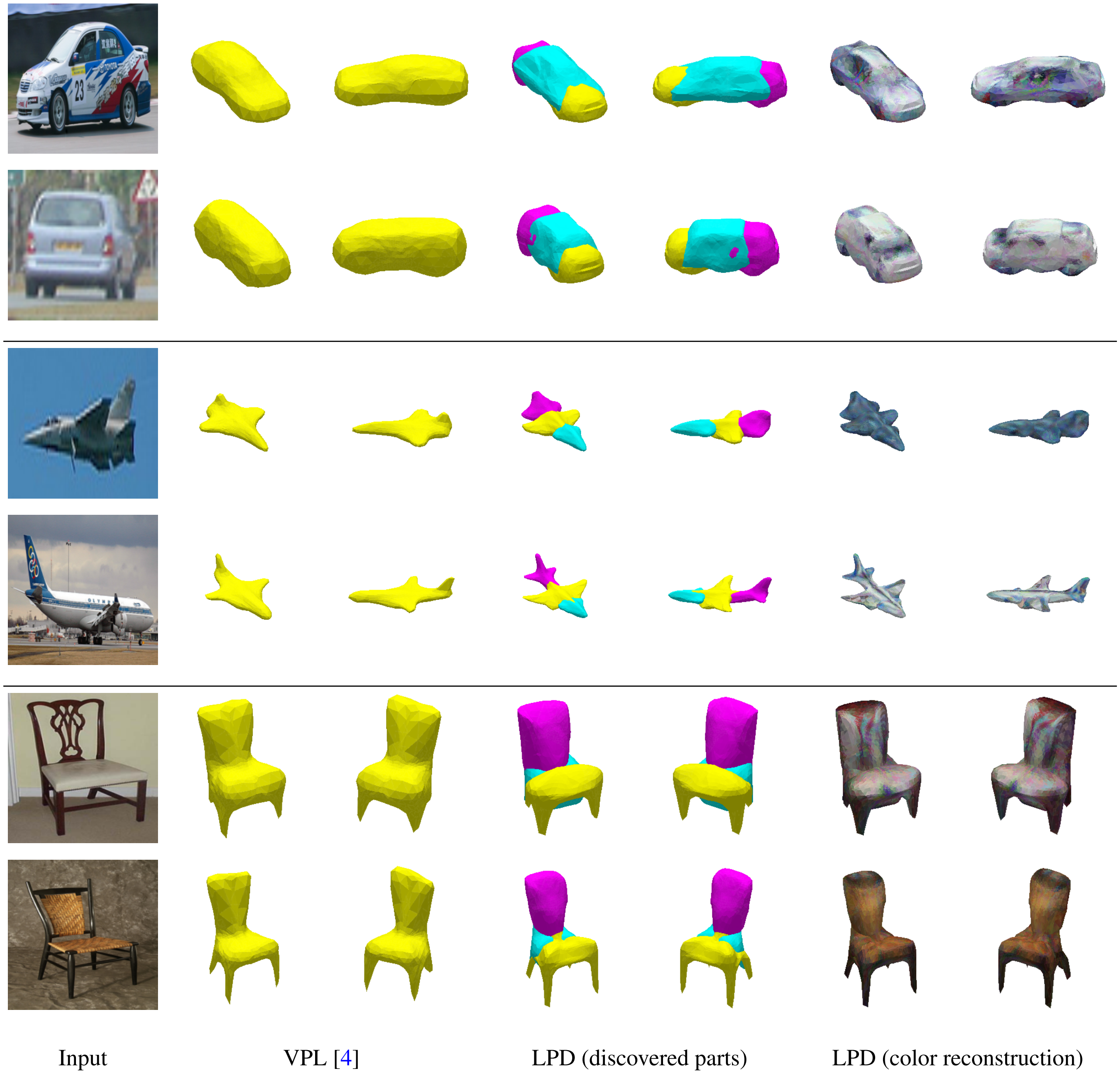

Figure 14: **3-part reconstruction results on the Pascal 3D+ dataset.** Note that the dataset size is significantly smaller than ShapeNet and ground-truth silhouette and viewpoint annotations are noisy. However, our model can still discover meaningful object parts on the challenging real-world images and capture more faithful details compared to the single-mesh models.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 1

[2] 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, 5

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 1

[4] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *CVPR*, pages 9778–9787, 2019. 3, 4, 9

[5] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. In *CVPR*, pages 3659–3667, 2016. 9

[6] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *CVPR*, pages 7708–7717, 2019. 3, 4

[7] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605, 2008. 3

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*, pages 8024–8035, 2019. 1

[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 9

[10] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, pages 75–82, 2014. 9

[11] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018. 1