Supplement for MVImgNet

A. Per-category Data Distribution

The category taxonomy is shown in Fig. I for MVImgNet, and Fig. II for MVPNet. The per-category data distribution is illustrated in Fig. III for MVImgNet, and Fig. IV for MVPNet. The average size is 921 per class for MVImgNet and 581 per class for MVPNet.

B. More Visualizations of Data Samples

MVImgNet. Fig. V presents a larger set of examples in MVImgNet. Several multi-view images and the corresponding class label are illustrated for each sample. It clearly shows the differences between each view, and comprehensive categories in our dataset.

MVPNet. Fig. VI shows various 3D point clouds from MVPNet. It can be seen that each sample has a distinct texture, noise, and pose, indicating real-world signals.

C. More Visualizations of Qualitative Results

Radiance field reconstruction. We visualize more results of generalizable NeRF reconstruction in Fig. VII, where the MVImgNet-pretrained model performs consistently much better than the train-from-scratch model.

View-consistent SOD. Fig. VIII illustrates more results of the view-consistent salient objection detection (SOD) task on our MVImgNet test set, where finetuning U2Net [74] on MVImgNet gains better result than the original U2Net.

D. More Experiments of Data Scalability

As indicated in the main paper, more power can be gained with more data utilized from our datasets. In this section, we provide more experimental results following such rules.

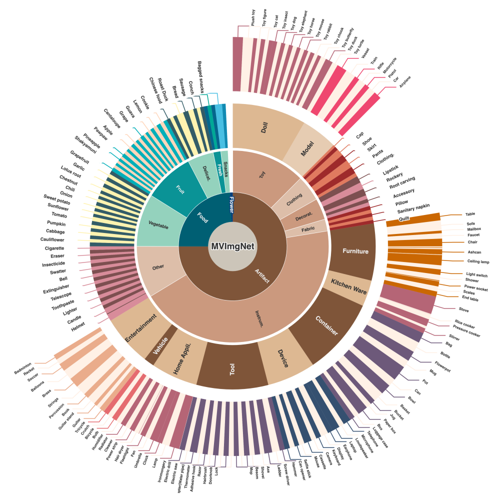

Figure I. **Category taxonomy of MVImgNet**, where the angle of each class denotes its actual data proportion. **Interior**: Parent class. **Exterior**: Children class.

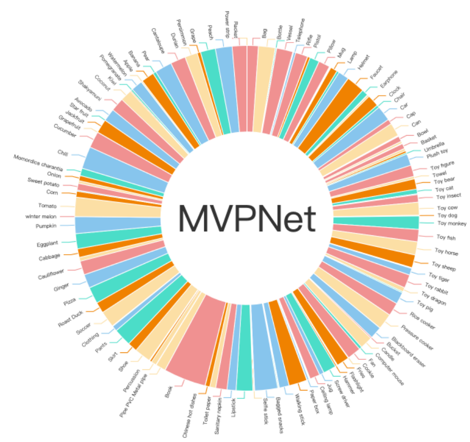

Figure II. **Category distribution of MVPNet.**

Multi-view stereo. Tab. I lists the MVS depth map accuracy on DTU [2] evaluation set. It shows that using larger amounts of videos from MVImgNet for pretraining yields higher accuracy.

View-consistent image classification. Similar conclusions are also found in the view-consistent image classification task. We progressively add more MVImgNet training data into MVI-Mix data (mixing the original ImageNet [24] data with MVImgNet data as stated in the main paper) to train ResNet-50 [45] and evaluate on MVImgNet test set. Tab. II demonstrates that adding more MVImgNet training data brings better view consistency for the image recognition task.

Real-world point cloud classification. Besides, as shown in Tab. III and Tab. IV, when employing larger ratios of

Method	$2mm\uparrow$	$4mm\uparrow$	$8mm\uparrow$
pretrained with 10k videos	52.96	72.25	83.79
pretrained with 50k videos	56.86	73.79	83.42
pretrained with 100k videos	58.63	75.20	84.28

Table I. **MVS depth map accuracy** on DTU [2] evaluation set, using **different amounts (10k, 50k, 100k) of videos** (one video may contain several multi-view images / frames) from MVImgNet for pretraining.

Scale	Confidence Var	Accuracy
ImageNet-only	0.207	53.09%
MVI-Mix with 20k videos	0.119	75.03%
MVI-Mix with 40k videos	0.114	76.88%
MVI-Mix with 80k videos	0.104	77.03%
MVI-Mix with 100k videos	0.102	77.31%
MVI-Mix with 120k videos	0.101	77.47%

Table II. **View-consistency image classification results** on MVImgNet test set, using **different amounts (20k, 40k, 80k, 100k, 120k) of videos** (one video may contain several multi-view images / frames) from MVImgNet for training ResNet-50 [45] (smaller Confidence Var and higher Accuracy indicate better view consistency).

data from MVPNet for pretraining both supervised (*i.e.*, PointNet++ [73], CurveNet [98]) and self-supervised models (*i.e.*, PointMAE [68]), the better performance can be achieved when fine-tuning them on ScanObjectNN dataset [89] for real-world point cloud classification task.

E. More Discussions about Our Datasets

Data filter. Our $\sim$219k videos are screened from $\sim$260k raw videos, where the videos with bad camera estimations are filtered. When building MVPNet, we select 90k (the most common 150 categories are chosen) videos, yielding 87k point clouds to remain after the manual cleaning.

Real-world captures. Note that when we capture the object videos, we maintain the *original* status of objects in *real-world* environments, *i.e.*, objects will *not be intentionally* displayed standalone for ideal 360° captures (*e.g.*, the sofa is against the wall). By doing so: **1)** The capture is easy to conduct, making it possible to build a very large-scale dataset. **2)** The produced data better matches the *real-world applications*, *e.g.*, our obtained point clouds are usually of partial views which are more like real-captured. **3)** The produced images usually contain the diverse scene-level *background*, instead of the 360° capture of single objects on a *clean* supporter. This better provides the potential for *in-the-wild* scene-level visual tasks.

Method	Add Random Rotation			
	from scratch	**25%**	**50%**	**100%**
PointNet++ [73]	76.50 / 73.42	77.82 / 75.98	78.11 / 76.13	78.76/76.54
CurveNet [98]	73.96 / 69.96	73.75 / 69.86	75.83 / 72.48	78.99 / 76.59
PointMAE [68]	83.17 / 80.75	83.83 / 81.94	85.22 / 83.34	86.19 / 84.60

Table III. **ScanObjectNN [89] real-world point cloud classification results** of using **different ratio (25%, 50%, 100%) of data from MVPNet for pretraining** under the setting of Add Random Rotation. The metric is **overall / average accuracy**.

Method	PB_T50_RS			
	from scratch	**25%**	**50%**	**100%**
PointNet++ [73]	78.80 / 75.70	79.67 / 76.63	81.36 / 79.33	80.22 / 76.91
CurveNet [98]	74.27 / 69.43	77.26 / 72.65	81.32 / 78.03	83.68 / 81.17
PointMAE [68]	77.34 / 73.52	82.75 / 79.90	84.18 / 81.41	84.13 / 81.92

Table IV. **ScanObjectNN [89] real-world point cloud classification results** of using **different ratio (25%, 50%, 100%) of data from MVPNet for pretraining** under the setting of PB_T50_RS. The metric is **overall / average accuracy**.)

F. Implementation Details

F.1. 3D Reconstruction

Radiance field reconstruction. We choose IBRNet [94] as the baseline method, and use the original training datasets of IBRNet [94], which include Google Scanned Objects [27], RealEstate10K [116], the Spaces dataset [29], and 102 real scenes from handheld cellphone captures. We pretrain IBRNet on the full MVImgNet dataset and finetune on the aforementioned IBRNet training datasets for 10k iterations. For each object, 8$\sim$12 views are used for training and 10 views for inference. #views is independent on #objects. The raw input resolution of each sample is used for computing, and it varies. The finetuning takes 10k iterations, and the scratch model is exactly the same as the author-released IBRNet model for a fair comparison. The pretraining takes about 3 days on 8 RTX3090 GPUs.

Multi-view stereo. Multi-view stereo (MVS) aims at recovering 3D scenes from multi-view images and calibrated cameras. As for the data preprocessing, 200K frames are randomly sampled from 100K videos in MVImgNet, and are resized to 640×360 or 360×640. We choose JDACS [103] to perform self-supervised pretraining on MVImgNet. JDACS takes multi-view images and corresponding poses as input, and uses MVSNet as the backbone to output the synthetic/pseudo depth, where the self-supervision signal is provided by multi-view consistency.

F.2. View-consistent Image Understanding

View-consistent image classification.

As mentioned in the main paper, we mix MVImgNet and original ImageNet [24] for creating a new training set. The hybrid datasets contain 1,100 categories (after removing the overlapping classes), coming from 500k frames of 100k MVImgNet videos and 200k ImageNet images.

View-consistent contrastive learning. We follow the original MoCo v2 to conduct experiments. For reducing view redundancy, we randomly sample 5 frames of each video from MVImgNet for finetuning. For each iteration, we randomly sample two view images from the same video as positive pair and apply random data augmentation to increase the generalization capability of the model, images from other videos will be treated as negative pairs

View-consistent SOD. We propose to leverage the multi-view consistency to improve SOD with the help of *optical flows*. The two adjacent frames should be the same after warping the optical flow to one of the other frames, yielding the loss of the optical flow as:

$$Loss_{OF} = \mathcal{M}(f_i) - \mathcal{M}(f_{i-1}) \cdot \mathcal{F}(f_i), \tag{1}$$

where i denotes the frame index, $\mathcal{M}$ means the mask, and $\mathcal{F}$ is the optical flow between f_i and f_{i-1} calculated before training. By adding $Loss_{of}$ into the original SOD loss, the final loss is:

$$Loss = \tau * Loss_{OF} + (1 - \tau) * Loss_{SOD}, \tag{2}$$

where τ is set to 0.15 in our experiments.

For fast training, we sample 10 frames uniformly from each video of 100k MVImgNet and 10, 553 training images from DUTS-TR [93].

F.3. 3D Understanding

All the experiments in 3D understanding strictly follow the original settings of the selected backbone networks.

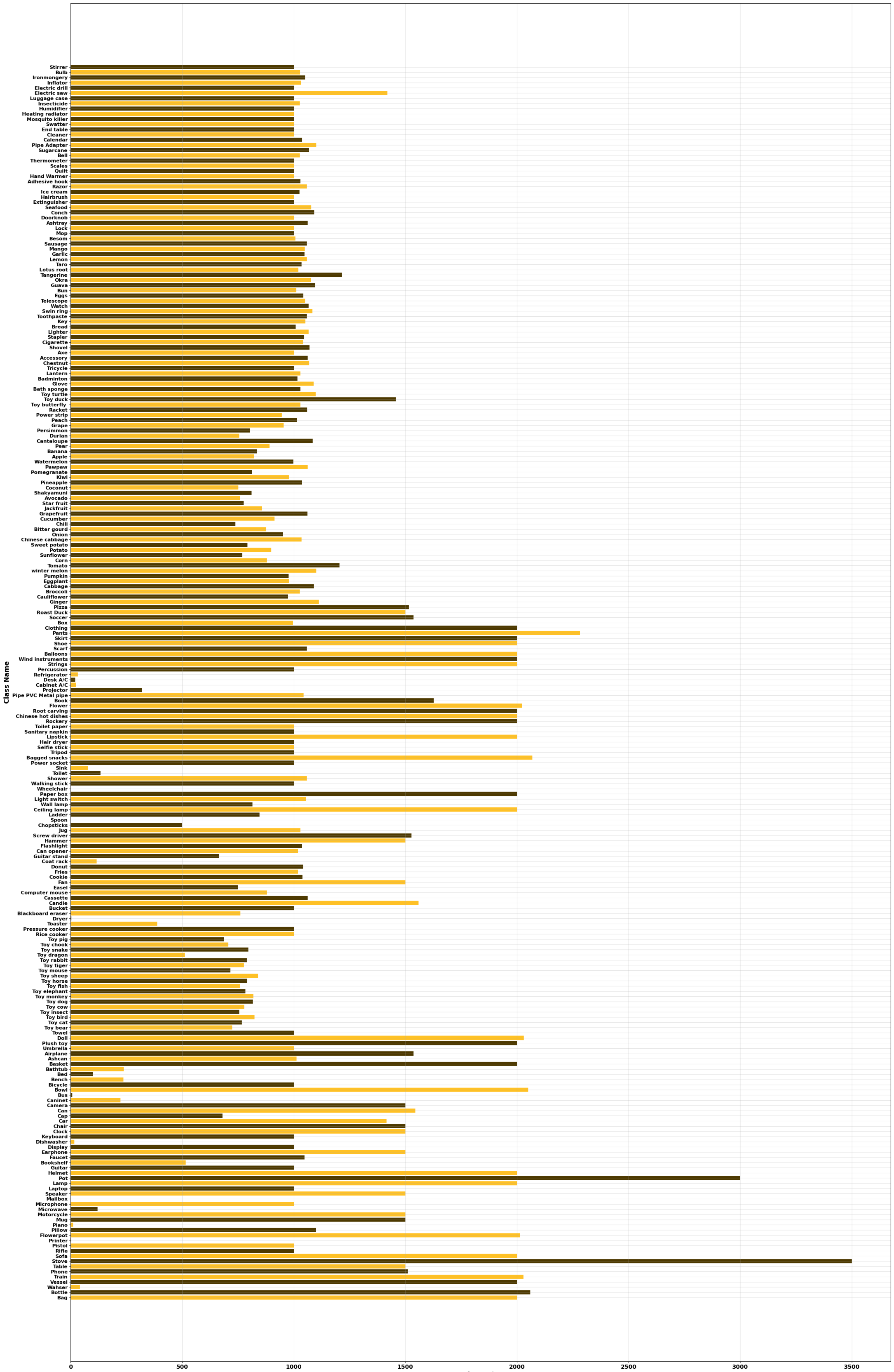

Figure III. Data amount of each category of **MVImgNet**.

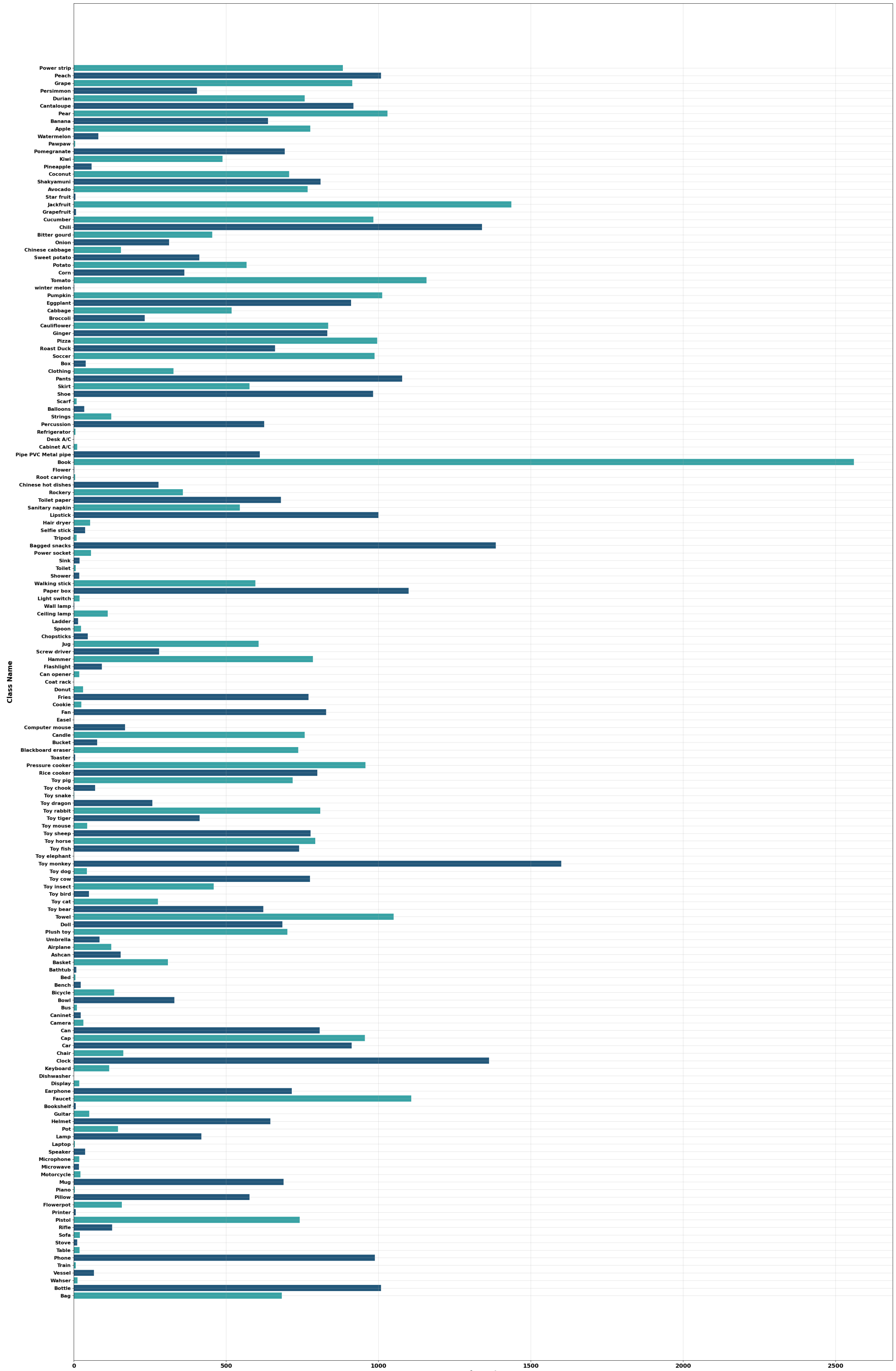

Figure IV. Data amount of each category in **MVPNet**.

Figure V. A variety of multi-view images in **MVImgNet**.

Figure VI. A variety of 3D object point clouds in **MVPNet**.

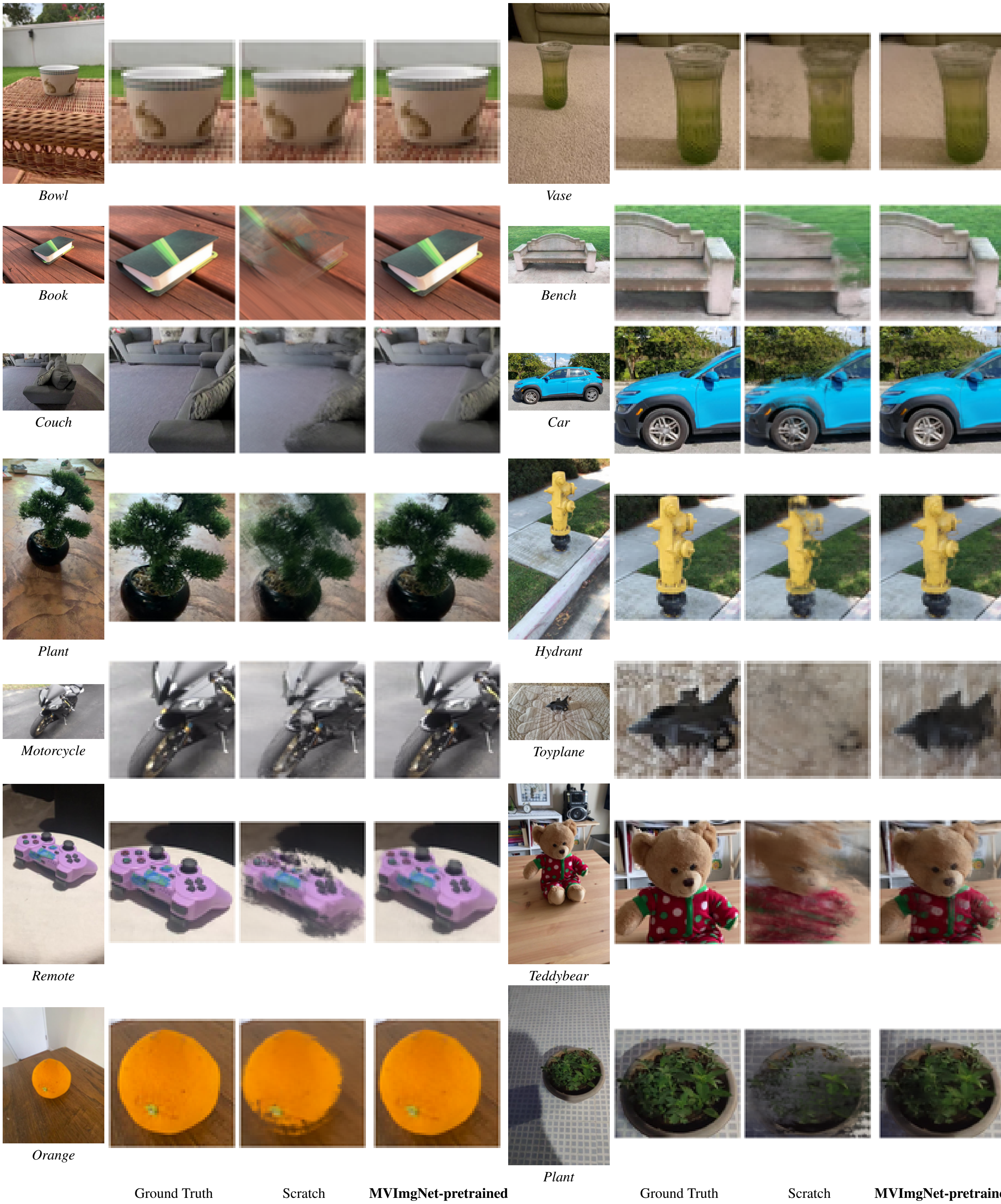

Figure VII. More qualitative comparison on real-world 360° objects [75] of **MVImgNet-pretrained** IBRNet [94] model and the **train-from-scratch** model.

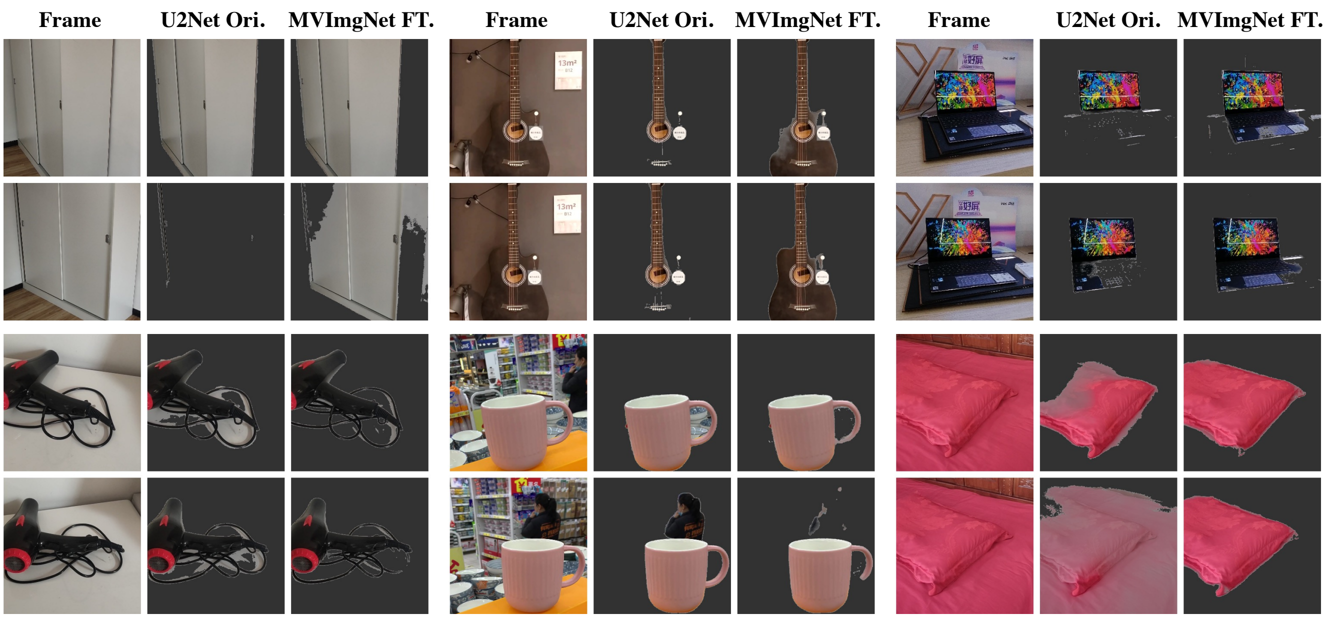

Figure VIII. More qualitative results of view-consistent salient object detection. **Finetuning U2Net [74] on MVImgNet** improves the performance.