Supplementary Material for Single Image 3D Interpreter Network

Jiajun Wu[1*], Tianfan Xue[1*], Joseph J. Lim[1,2], Yuandong Tian[3], Joshua B. Tenenbaum[1], Antonio Torralba[1], and William T. Freeman[1,4]

[1]Massachusetts Institute of Technology
[2]Stanford University [3]Facebook AI Research [4]Google Research

In this supplementary material, we provide (1) more results on Keypoint-5 chair and sofa, (2) a quantitative comparison between 3D-INN trained using the paradigm described in Section 3.3 in the main submission, and a simplified version trained end-to-end with real images only, and (3) detailed parameters of 3D-INN. We also show 3D rendering and image retrieval results in attached videos (rendering.mp4 and retrieval.mp4).

1 Results on Keypoint-5 Chair and Sofa

We here supply more results on chairs and sofas in Keypoint-5. Figures 1 and 2 show the estimated skeletons for chairs and sofas, respectively. Images are randomly sampled from the test set.

2 Validating Training Paradigms

As mentioned in Section 3.3 in the main text, here we show quantitative comparisons between 3D-INN trained using our proposed paradigm, and the same network but only end-to-end trained with real images, without having the two pre-training stages. We called it the *scratch* model.

As shown in Figure 3, 3D-INN performs much better than *scratch*. The average recall of 3D-INN is about 20% higher than *scratch* in 3D structure estimation, and about 40% higher in 3D pose estimation. This shows the effectiveness of the proposed training paradigm.

3 Network Parameters

As mentioned in the main text, our network consists of three components: first, a keypoint estimator, which localizes 2D keypoints of objects from 2D images by regressing to their heatmaps (Figure 4a and b); second, a 3D interpreter, which infers internal 3D structural and viewpoint parameters from the heatmaps (Figure 4c); third, a projection layer, mapping 3D object to 2D keypoint locations so

∗ indicates equal contributions.

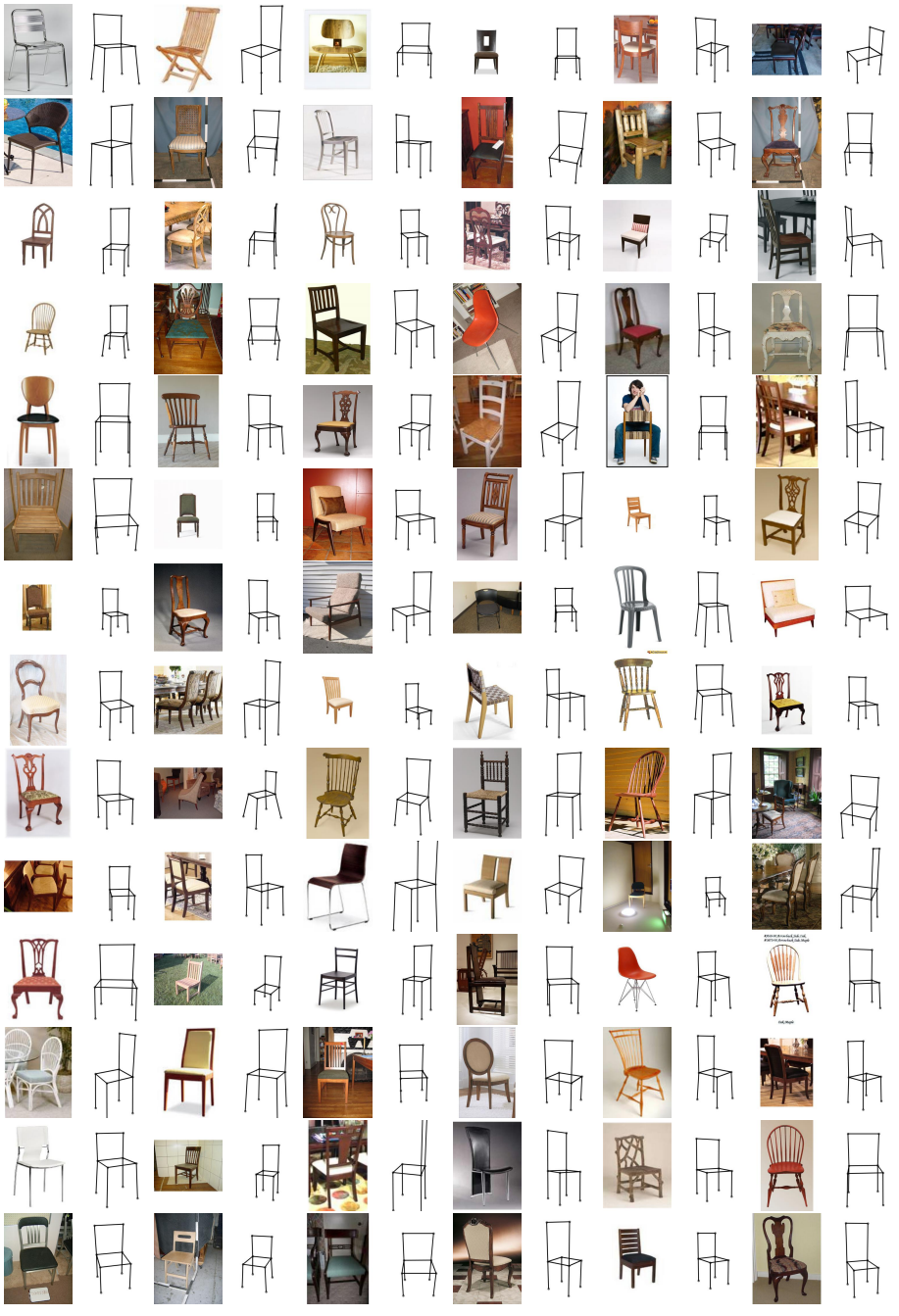
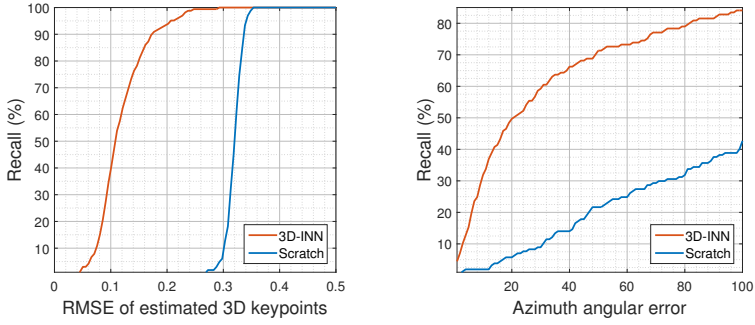


Fig. 1. Estimated 3D skeletons on more Keypoint-5 chair images. Images are randomly sampled from the test set.

Fig. 2. Estimated 3D skeletons on more Keypoint-5 sofa images. Images are randomly sampled from the test set.



(a) Structure estimation (b) Pose estimation

Fig. 3. Evaluation on chairs in the IKEA dataset [1]. The trained network using our paradigm (3D-INN) is significantly better than *scratch* on both structure and pose estimation.

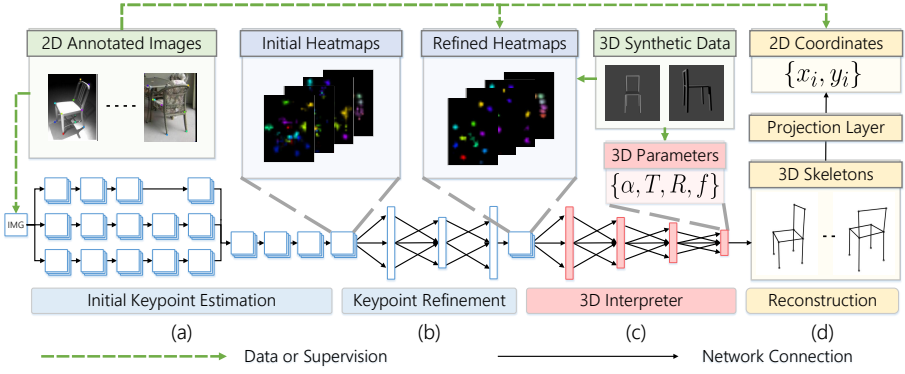


Fig. 4. 3D-INN takes a single image as input and reconstructs the detailed 3D structure of the object in the image (*e.g.*, human, chair, *etc.*). The network is trained independently for each category, and here we use chairs as an example. **(a)** Estimating 2D keypoint heatmaps with a multi-scale CNN. **(b)** Refining keypoint locations by considering the structural constraints between keypoints. This is implicitly enforced with an information bottleneck which yields cleaner heatmaps. **(c)** Recovered 3D structural and camera parameters $\{\alpha, T, R, f\}$. **(d)** The projection layer maps reconstructed 3D skeletons back to 2D keypoint coordinates.

that real 2D-annotated images can be used as supervision (Figure 4d). Specifically, the keypoint estimator can further be divided into two parts, the first for initial estimation (Figure 4a) and the second for refinement (Figure 4b).

Our network for initial keypoint estimation (part a) is based on the network proposed by Tompson *et al.* [2]. The network takes multi-scaled images as input and estimates keypoint heatmaps. Specifically, we apply Local Contrast Normalization (LCN) on each image, and then scale it to 320×240, 160×120, and 80×60 as input to three separate scales of the network. The output is k

heatmaps, each with resolution 40×30, where k is the number of keypoints of the object in the image.

At each scale, the network has three sets of 5×5 convolutional layers with 128 filters (with zero padding), each of which is followed by a ReLU layer and 2×2 pooling layers. Those are followed by a 9×9 convolutional and a ReLU layer, with 512 filters. The final outputs for the three scales are therefore images with resolution 40×30, 20×15, and 10×7, respectively. We upsample the outputs of the last two scales to ensure they have the same resolution as the first scale (40×30). The outputs from the three scales are later summed up in a filter-wise manner, and sent to a Batch Normalization layer, followed by three 1×1 convolution layers, whose goal is to regress to target heatmaps. The number of filters in these 1×1 convolutional layers is 512, 512, and k, respectively, where k is the number of keypoints. We found that the batch normalization layer we added is critical for convergence, while Spatial Dropout, proposed in [2], does not affect performance.

For keypoint refinement (part b), we use three fully connected layers with widths $8,192$, $4,096$, and $8,192$, respectively.

The 3D interpreter (part c) contains four fully connected layers as our 3D interpreter, with widths $2,048$, 512, 128, and $|S|$, respectively, where $|S|$ is the number of parameters we estimate.

The projection layer (part d) is just a single layer with no learned parameters, calculating projected 2D keypoint locations from estimated 3D skeletons using the Equation (2) in the main submission.

4 3D rendering and image retrieval

Please refer to the attached videos (rendering.mp4 and retrieval.mp4) for 3D rendering and image retrieval results.

References

1. Lim, J.J., Pirsiavash, H., Torralba, A.: Parsing ikea objects: Fine pose estimation. In: ICCV (2013) 4
2. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: CVPR (2015) 4, 5